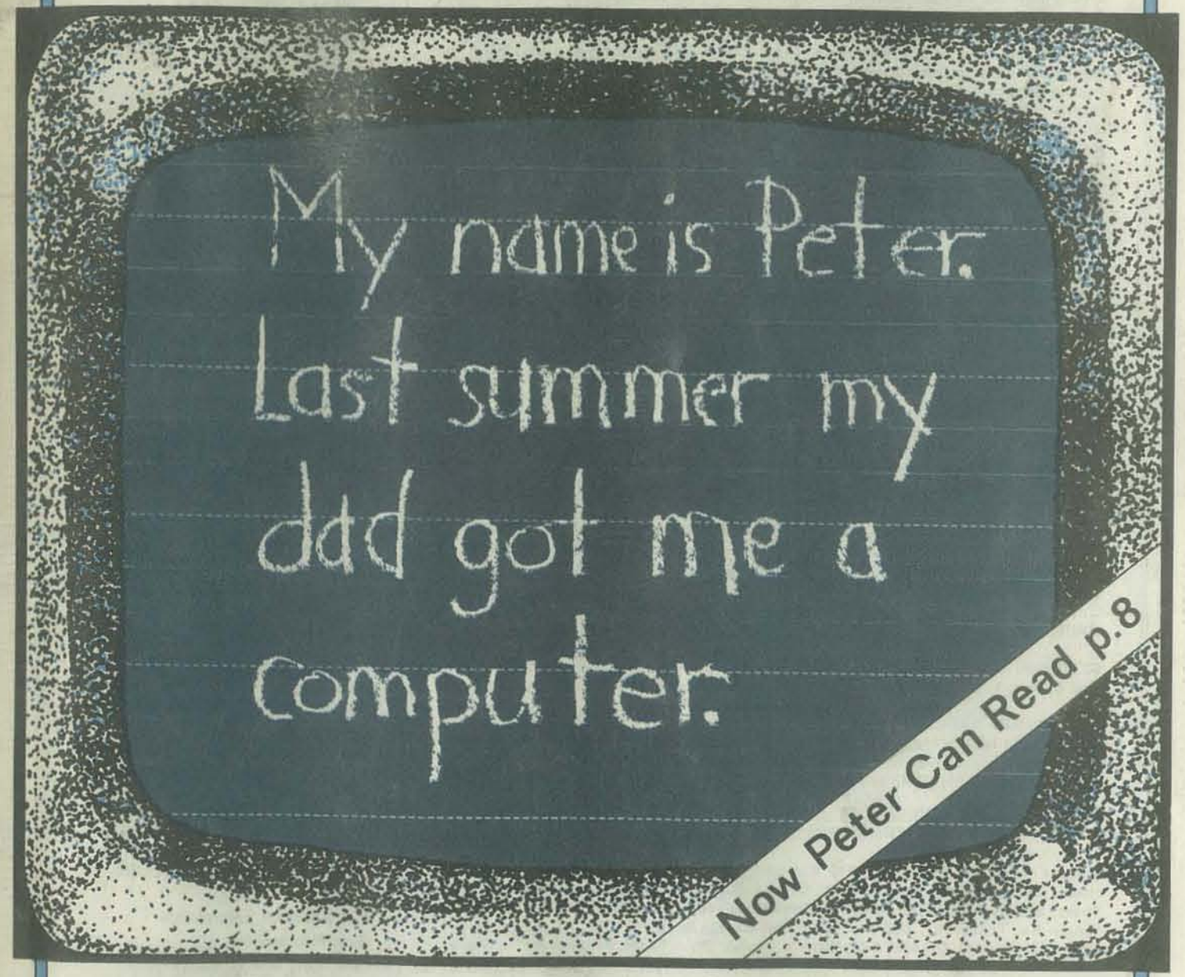
Second Class Postage Paid at Menlo Park, CA
Address Correction Requested

FORMERLY PEOPLE'S COMPUTERS

Recreational COMPUTING

VOL 7 NO 6 ISSUE 39 MAY – JUNE 1979 $2.00



Apple · PET · Reviews · APL · Games · TRS-80 · Programs
PASCAL · Futureplay™ · BASIC · Reviews · FRP · SWTPC

SUBMITTING ITEMS FOR PUBLICATION

LABEL everything with your name, address and the *date*; tapes should also include the program name, language and system. TYPE text if at all possible, double-spaced, on 8½x 11 inch white paper. DRAWINGS should be as clear and neat as possible in black ink on white paper.

LISTINGS are hard to reproduce clearly, so please note:

- Use a new ribbon on plain white paper when making a listing; we prefer roll paper or fan-fold paper.
- Send copies of one or more RUNs of your program, to verify that it runs and to provide a sense of how things work—and to motivate more of us to read the code. RUNs should illustrate the main purpose and operation of your program as clearly as possible. Bells, whistles and special features should just be described jn the documentation unless they're particularly relevant.
- Make sure your code is well documented—use a separate sheet of paper. Refer to portions of code by line number or label or address, please, not by page number. When writing documentation, keep in mind that readers will include beginners and people who may be relatively inexperienced with the language you're using. Helpful documentation / annotation can make your code useful to more people. Documentation should discuss just which cases are covered and which aren't.
- If you send us a program to publish, we reserve the right to annotate it (don't worry, we won't publish it if we don't like it).
- Last but not least, please try to limit the width of your listings: 50-60 characters is ideal. Narrow widths mean less reduction, better readability and better use of space.

LETTERS are always welcome; we assume it's OK to publish them unless you ask us not to. Upon request we will withhold your name from a published letter, but we will not publish correspondence sent to us anonymously. We reserve the right to edit letters for purposes of clarity and brevity.

SUBSCRIPTION INFORMATION

U. S. RATES

- ☐ $10/1 yr. (6 issues)
- ☐ Retaining subscription @ $25 ($15 tax deductible)
- ☐ Sustaining subscription @ $100+ ($90+ tax deductible)

FOREIGN RATES

Payments must be in $US drawn on a US bank.

- ☐ $17 Canada First Class
- ☐ $23 Rest of World Airmail
- ☐ $14 World Surface Mail

Please allow 6-9 weeks for your first issue to arrive.

Delivery of foreign mail is slow and unreliable. We strongly advise airmail.

BACK ISSUES

$2.50 each: Vol. 6, Nos. 1, 2, 3, 4, 5; Vol. 7, Nos. 1, 2

$3.00 each: Vol. 7, Nos. 3, 4, 5

Outside the U.S. add $.50 per issue.

EXPIRATION DATE: To decipher the expiration date of your subscription, look at the top right hand corner of your address label. The last two digits refer to a code/issue number that is your expiration date. Hence, read 38 as 3/79, 39 as 5/79, 40 as 7/79, 41 as 9/79, 42 as 11/79, 43 as 1/80, and so on.

FOREIGN DISTRIBUTORS OF *RECREATIONAL COMPUTING*

Vincent Coen, **LP Enterprises**, 313 Kingston Road, Ilford IG1 1PJ, Essex, UK; Rudi Hoess, **Electronic Concepts PTY Ltd.**, Ground Floor Cambridge House, 52-58 Clarence Street, Sydney NSW 2000; **ASCII Publishing**, 305 HI TORIO, 5-6-7 Minami Aoyama, Minato-Ku, Tokyo 107, JAPAN; *Eastern Canada:* Liz Janik, **RS-232**, 186 Queen St. W., Suite 232, Toronto, ON M5V 1Z1; *Western Canada:* Brian Wiebe, **Kitronic Audio, Digital & Electronic Components**, 26236 26th Ave. RR5, Aldergrove BC VOX 1A0, Canada.

Printed by Nowels, Menlo Park, CA

Recreational Computing (USPS 356870) is published bimonthly by People's Computer Company, 1263 El Camino Real, Box E, Menlo Park, CA 94025. People's Computer Company is a tax-exempt, independent, non-profit corporation, and donations are tax-deductible. Second class postage paid at Menlo Park, California, and additional entry points.

Recreational COMPUTING

Volume 7 Number 6
May – June 1979

formerly **people's computers**

p. 8

p. 16

p. 48

p. 34

COVER
Ann Miya & Aleeca Harrison

Special Features

Articles

Games & Stuff

Departments

Editors' Notes

This issue is packed! Packed with software, software reviews, places to buy software and places to get free software. There are listings in BASIC, PASCAL and APL. There are programs for the PET, TRS-80, Apple, SWTPC and the SOL. The announcements section has a different look. Two new (reincarnated?) departments—"Futureplay™" and "Programmer's Toolbox™"—begin (again) with this issue. There is information ("Dragonsmoke") and an introductory article ("What Is All This Stuff?") on fantasy gaming. FORTRAN Man and Billy BASIC are back together helping Linea.

The lead article, "Testimony to a Micro—Peter Can Now Read," is an inspiring documentation of how technology is being used as it makes its way into the hands of "the people." The lowly Level I TRS-80 has been used to perform a miracle as far as Peter's parents are concerned. A miracle that they helped implement.

And then, as always, there is the delightful Letters section. We get letters from supporters, critics, friends, people in need of information, kids, parents, teachers and lawyers. Our letter writing community is a unique group of people.

What more could you ask from a mere 64-page magazine? Well there's "Golf Handicapping" and "Concept Sans Computer" and . . . but get busy and read and read and read and . . .

Ramon Zamora
Louise Burton
Bob Albrecht

CORRECTION

bes-ti-ary/ 'bes(h)-chē,er-ē/ n [ML *bestiarium*, fr. L, neut. of *bestiarius* of beasts, fr. *bestia*]: a medieval allegorical or moralizing work on the appearance and habits of animals.

The Be(a)stiary
(Mar.-Apr. issue, 1979)

CORRECTION

Two long-time sustaining/retaining subscribers, Bill Godbout Electronics and Algorithmics, Inc., were not listed as such in several of the past few issues. We wish to apologize to them and as a partial penance, we will write their companies' names below as many times as we can. — The Editors

BILLGODBOUT ELECTRONICS& ALGORITHMICS,INC. BILLGODBOUT ELECTRONICS & ALGORITHMICS,INC. BILL GODBOUT

Letters

HELP! HELP!

I teach 5th grade, and I'm trying to convince my school board that computers in an elementary classroom are not a $2,000 joke. I need help! I'd appreciate any information anyone could send me on references on the use of computers in elementary classrooms—most of it seems to deal with high school. I am also interested in finding sources of programs appropriate for elementary drill and instruction.

I've got *What To Do After You Hit Return* and can't wait to get something to run those programs on for my students!

Glenn Fisher
Armstrong School
2849 Calais
San Ramon, CA 94583

OK, all you teachers out there who've already won this battle, write to Glenn and share your strategy, source materials, clinching arguments. —Ed.

NOVELIST IN SEARCH OF CO-AUTHOR (COMPUTER ONLY)

I am currently writing a novel. My friend Kent, a painter, will then do some drawings/paintings that will be lithoed.

Now, I would like to put those two items through a computer, and I understand that there are roughly two ways of doing it:

First, an IBM 370 married to a CAPAM system. I'm not sure about that. Or, a system roughly equivalent to the NASA 2 model with a visual subsystem made by GE. This was used by Peter Kamnitzer to make a film called *City Scape 1970.*

Anyway, the software I am interested in, the process or programming, is to allow the computer to respond in its way to the material, rather than simply using the computer as a large version of a paint brush or typewriter.

Letters Letters Letters

Do you have any ideas/suggestions to make? I am very open now.

Peter Sorgen
Box 9932
537 Jones Street
San Francisco, CA 94102

Any readers with experience in creative text-editing? Please send your suggestions to Peter.

'CONCEPT' RE-CONCEIVED

I have enjoyed your articles, reviews, and programs in *Recreational Computing.* The game of Concept was quite interesting. Here is a list of changes to show the output a little better. They are yours to use, publish or whatever.

Delete the following:

```
130
300
410
1000-1020
1600-1630
```

Add the following:

```
118 DIM L$(7)
305 Z=0
417 GOSUB 1700
420 PRINT Z;:FOR K=1 TO 7:
    PRINT L$(K);:NEXT
425 PRINT
1700 FOR K=1 TO 7
1710 IF D(K)=0 THEN L$(K)=
    "0" ELSE L$(K)="1"
1720 NEXT
1730 RETURN
```

The sub-routine has the effect of producing leading zeros in the valid lines. Give it a try. I'd be pleased to hear from you.

R. L. Wagner
9035 Niles Center Road
Skokie, IL 60076

Thanks for your INPUT! Look at Eryk Vershen's Concept solutions in this issue also. —RZ

ABOUT THOSE 'WORTHWHILE' THINGS COMPUTERS DO . . .

What a relief!

In reply to Q. Burke's letter (Jan-Feb issue) complaining of the overabundance of games, I'd like to point out a few things about computers in general.

Computers never have, cannot, and possibly never will be able to do your thinking for you. If a human achieves a thorough understanding of a problem, that human can write a program which "freezes" the thought patterns necessary to its solution (given a defined set of input parameters). Modification of this pattern, however slight, requires another human who thoroughly understands:

the original problem,
the previous human's method of solution,
the previous human's programming style,
the new objective,
the changes necessary.

So much for the creative part. The reason that all this pleasure/pain of programming is done at all divides into two broad categories. The first is that the problem is so boring that any normal specimen of H. Sapiens is likely to fall asleep on the job. Example: maintaining a mailing list of more than about 100 names (anything less is easier to keep alive on an office copier). The second reason is that in a business, the person who understands a recurring problem is often too busy to solve it at each occurrence. Example: putting a new person on the payroll (. . . but sir, you have to have a Social Security Number . . .).

Note that the original users of computin' engines—engineers and scientists—are largely overlooked. They're back in the lab, playing with their toys.

Where is our freedom?

Let a computer do my income tax? The I.R.S. will do it for free—on a computer, yet! They'll even crunch the data several different ways, and pick the way that results in the least tax. What they *won't* do, and this is where accountants (not computers) come in, is restructure your books to (1) minimize taxes, and (2) give you a better view of your financial picture. Never let a computer (*or* an accountant) manage your finances. You spend the money. They keep track.

As for setting up a corporation, a computer is about as appropriate as a screwdriver. A corporation is a legal entity. Go to a lawyer, or do it yourself if you must (not advisable). And who says small business doesn't get the breaks? They have the greatest flexibility, the least regulation, the best people, the lowest overhead and the most fun. They don't need the nightmare of a computerized organization. (Ask any corporation's management what *they* think of the computer center.)

It is my considered opinion that the engineers and scientists back in the lab are experiencing the most freedom. They're the ones who are capable of using the computer at its maximum capacity, its highest efficiency, and if the hardware isn't designed the way they like it, they're the ones who can (and probably will) change it. Now go back there and dump a directory in their files.

Hmm . . . Star Trek.

I rest my case.

Ralph McElroy, Publisher
CLOAD Magazine
Box 1267
Goleta, CA 93017

'UPSET DRAGON' UPSETS READER

Dear Dragon,

Page 51 of the March-April issue of *RC* shows the following BASIC program:

```
100 IF X = 3 THEN Y = Y + 1 : Z = Z + 1
110 IF X<>3 THEN Y = Y - 1 : Z = Z - 1
```

Letters Letters Letters

Good grief! Who would write a BASIC program like that? In any reasonable BASIC (e.g. Applesoft) one would simply write the following line:

```
100 S = 2↑(X = 3) - 1 : Y = Y + S : Z = Z + S
```

Jim Day
17042 Gunther Street
Granada Hills, CA 91344

A DIFFERENT WAY TO 'FLOAT'

I read with interest Mark Zimmerman's article, "Snooping with Your PET," which contained floating point binary instructions (PC, Sept-Oct 1978).

Here is a slightly different version of your Program B:

```
10 V = 8194 - FRE (0)
20 INPUT X
30 N = N+7: Z = V: POKE 830,N
40 PRINT PEEK (Z); PEEK (Z+1); PEEK (Z+2);
   PEEK (Z+3); PEEK (Z+4)
50 END
```

Hope to see more of Mr. Zimmerman's articles.

Bob Feniger
1081 B Treat Ave.
San Francisco, CA 94110

GETTING IT ALL TOGETHER IN FAIRFIELD, CALIFORNIA

Don't mind your change in name as long as the information keeps coming! Since the loss of *ROM*, yours is the only magazine that deals well with the philosophic and basic issues of computing. Keep those magazines coming!

I am a member of a non-profit organization in the Fairfield, California, area (halfway between San Francisco and Sacramento) called Synergetic Systems Ultd. We formed our organization way back in 1972 to assist local residents in learning how to use—and not be used by—high technological tools. The tool we were involved with at the time was the Video Port-a-Pac. Since that time we have moved into the area of computers and are presently planning a project with some similarities to the Community Memory Project and the Marin Computer Center. We hope to provide our community with a place where people can come to learn about the community, themselves, computers and other areas of interest.

For example: People who move into the area would hear from advertising (Welcome Wagon, word-of-mouth, etc.) that there is a place (storefront) in the community where they can find just about any kind of information they might need about the community and its resources, from the names of local doctors to listings of job opportunities and entertainment. To connect with this information, they would use a data management system (perhaps similar to "WHATSIT") running in a microcomputer. Not only would they be able to receive the information they need, but they would become *aware* of their potential when assisted by the use of the computer and other tools.

They will see others utilizing, enjoying, learning and controlling high technology tools: people playing Star Trek or Runequest, making video programs, learning to utilize cable television's "public access." For once they will be able to *talk back* to the boob tube. They will be able to learn how to program and *use* computers in their home or business—in an environment that is positive and non-threatening. If they are unable to find the information they need, the operators of the program will make every effort to find the answer and add it into the data base.

Citizens or government leaders will be able to run polls on local issues. (We are presently working on a simulation that will provide participants an experience of the dynamics that are involved in running a county government program.) There will be more involvement and sensitivity to the needs of the community by the leaders and those who elect them. Perhaps for once in their lives, people will find they are not hindered by technology but *assisted!* They will be able to decide for themselves which direction they might want to go and utilize tools that will assist them in their quest.

Imagine having one place in each major community that operates like the "gatekeeper" of earlier days. Many government agencies that provide specialized information would become obsolete. In our county, for example, people needing information regarding alcohol or drug abuse services can call over eight different agencies and unless it is the agency that will help their specific problem, they will all get another phone number to call. With our center, all this information would be under one roof and phone number. Just imagine how this could and would begin to eliminate duplicated services. Imagine having a referral service in which doctors, lawyers, dentists, counselors, accountants, every sort of business can list their services in a format to their liking, including what they do, why they do it, what the normal costs are, and any other information they may want the prospective client to know. Sure beats the yellow pages, doesn't it?

I know it sounds like utopian thinking, but I believe it is reachable (even if on a smaller scale than my dream). Currently we provide "hands-on" training to local youths and are involved in collecting "interest" and "skill" information from as many residents as we can to start a data base. We are currently writing a program that will assist us in determining what kinds of interests and skills are in the community, and after that we will start finding ways of connecting people with similar interests.

We would be more than happy to interact with others with similar ideas. If anyone can figure out a way to get "seed money" for a project like this, please contact us.

Tony Severa
Synergetic Systems Ultd.
131 Highland Ave.
Vacaville, CA 95688

Letters Letters Letters

LOOKING FOR A PORT-OF-CALL

I am trying to locate a board game called Port-of-Call in which players control Pacific Ocean shipping lines and engage in the transportation of goods between ports while battling time and the weather. I last played Port-of-Call about ten years ago. The game was well designed, and now I would like to develop a real-time simulation of it on a computer, using as many of the original game features as possible. Unfortunately, I have not been able to locate a copy of the game (I need to refresh my memory on exactly how it was played). Can your readers help me locate Port-of-Call?

Mike Gabrielson
Box 2692
Stanford, Calif. 94305

THEY LOVE US IN SAN DIEGO

I just received my first copy of *Recreational Computing*, and it is GREAT.

There are so many wonderful articles and games, it is difficult to decide where to start reading first.

I am delighted to have such a marvelous resource for ideas and activities to use in workshops, classrooms, and at home.

Thank you.

Jane Donnelly Gawronski
Curriculum Coordinator
Mathematics/Computer Extended Instruction
San Diego County Schools
San Diego, CA 92111

NO END TO APPLE MATH

I am now an owner of an Apple II computer. In using the Apple math program in your Sept-Oct 78 issue, written by John Gaines, I find there is no end to the program.

In line 1035, either you type "OK" and continue with more math, or you hit the return which also starts the program over—the same as typing "OK."

I modified his program as follows:

```
1035 INPUT "IF YOU WANT
     ANOTHER PROBLEM, TYPE
     'OK-NOK'",D$
1042 IF D$="NOK" THEN
     1055:REM NEW LINE
1065 PRINT:PRINT" ",
     " ","  APPLE!!":
     GOTO 8999
8999 END
```

This allows a person to say "NO" to more math problems and also to display the information in lines 1055 and 1065 which I did not get with the program as listed in your magazine.

Lloyd Dawson
RFD 2
Ossian, IN 46777

Looks like you got to the core (whoops!) . . . heart of the matter. — *Ed.*

AUTHOR REFINES 'MPG'

No program is ever complete . . . there is always something that can be added to make it run better. After reviewing the MPG program that you were kind enough to print in the Jan-Feb '79 issue, several changes came to mind to make it "better," and I suspect some of your readers will have additional suggestions.

First, the biggest error that I tend to make is placing the line numbers too close, a holdover from my Tiny BASIC days where the numbers could run only from 2 through 255. This creates a problem when you want to add a line or two to patch something.

Second, I found it somewhat aggravating to wait while the data tapes are being written or read, since there was no way to know how far along the computer had gotten.

Finally, I found that you can hang up the program if you accidentally enter an odometer reading that is less than the previous reading, since that yields a negative number and fouls up the calculations.

The last two problems are taken care of by the following lines, which you may want to pass along to your readers . . . the first problem is a matter of habit, which I will try to break.

```
28 INPUT "YOUR SELECTION";
   S: IF S>6 GOTO 28
29 ON S GOTO 50,80,160,210,
   110,130
```

Delete lines 30-35. *Not necessary, but looks better!*

```
56 FOR I=1 TO D:PRINT AT 576,
   "READING RECORD #";I
139 FOR I=1 TO D:PRINT AT 576,
   "WRITING RECORD #";I
```

Lets you know how many records have been read/written.

To create room for the following line, renumber lines 94 thru 97 to 95 thru 98. *Prevents using an odometer reading that is too low!*

```
94 IF A(D)<=A(D-1) D=D-1:
   CLS:PRINT"*** ERROR ***":
   GOTO 20
```

Milan D. Chepko, M.D.
119 Belleville Court
Thief River Falls, MN 56701

The MPG listing also contains two typos. In line 57 there is an "S" in INPUT #-1 and in line 95 the GOTO should be GOTO 97. — *Ed.*

THANKS FOR THE APPLE TURN-ON

It was with great pleasure that I opened the Jan-Feb *Recreational Computing* and found material on the Apple II. Using the information about the I/O control socket, together with the Softape listening program, it should be fairly easy and inexpensive to give voice commands for turning on the lights, etc.

Keep those Apple articles coming.

Winston Cope

a testimony to a micro

peter can now read

BY JOHN POLLARD, father of Peter
89 Bunarba Road, Gymea, N.S.W. 2227, Australia

This article is reprinted from a magazine published in Australia called COM-3. COM-3, *P. O. Box 268, Niddrie, Vic., Australia, is a non-profit magazine affiliated with the Computer Education Group of Victoria. Its purpose is to be a resource for those interested in the use of computers in education or personal computers. The name* COM-3 *is derived from the first three letters of the words COMPUTER COMMUNITY COMMUNICATIONS, and symbolizes the growing interaction between them.* COM-3 *is published 5 times per year at roughly two-month intervals.*

You will find the article to be one of the most exciting things that you will read this year. —*RZ*

the problem

Peter is a 10-year-old lad who just cannot read—he is intelligent, but dyslexic.
Words, like the end of a rainbow, are always out of reach—the letters "ss" in grass seem to force their way to the beginning of the word to give an almost indecipherable code. Peter is locked out from the beauty of the world of words. A trick of the brain makes fun of words by throwing letters about like a sorter at the post office. His parents have tried everything—additional help after school hours and special schooling for over a year by those skilled in the problem; but no, he cannot yet read. Then along comes a microcomputer and the lad can now read —a new world is open to Peter.

the gamble

Imagine the naked simplicity of the math problem 12 X 6 ? buried in a jungle of overlaying words: "1 dozen bananas are purchased at 6 cents each, what is the cost?"

Peter just could not grasp his arithmetic as words acted as a sentinel barring entry to the heart of the problem. Late in 1977 a programmable HP25 calculator was purchased by a scientifically minded father for his own use at work. Then in desperation about February the calculator was put to work to teach Peter arithmetic using computer (calculator) assisted instruction (CAI) techniques. Peter caught on—the simplicity of the little machine matched his need at the time and sums became alive. Three months later a small complete unit 4K, TRS-80 microcomputer appeared in a local Tandy store. Peter's mother had just received a cheque for $800.00 from an auntie and his father just happened to visit the store. A swap of money and machine took place. Both parents were willing to gamble that if a calculator can teach arithmetic then maybe a computer, albeit micro, could perhaps teach reading skills—but how?

the machine

By some reports (Tom Williams, *People's Computers*, Vol. 6, March-April 1978, for example) the TRS-80 and its Level I BASIC, like Peter, was labelled as a dummy. How could anyone even think of using the thing for CAI with almost non-existent character string manipulation features? And as for a keyboard not disconnecting from communicating to a monitor screen when the machine is busy thinking about something else—well, how dumb can you get? With this sort of challenge (thanks Tom) how could a person not respond—in fact perhaps the TRS-80 is the only machine off the shelf that can do the job reported in this work!

The apparently "unloved" TRS-80 found a welcome home with the Pollard family. Perhaps our sensitivity to Peter being labelled unmercifully made us look beyond the surface in our appraisal of the machine's capability. In any case, the little machine, lovingly called TRIS by the family, has proven itself in the way it has helped Peter. And Peter has proven himself in the way he has responded. No—neither of them are dummies; nothing could be further from the truth.

The Level I BASIC language available with the machine when purchased, although restrictive, has many nice features. It seems to be true in the author's experiences that anything goes — think of a way of doing something and the method will work. (Maybe this is purely an observation on the fact that it works because it is simple.) Certainly the language with its compactness of presentation makes good use of the small amount of memory, 4K bytes, purchased with the machine. This does not say that the Level II BASIC running with 16K of memory would not do the job more effectively, but the question of whether the additional cost of $500.00 puts the approach out of reach of most parents and schools needs be seriously considered.

Yes, TRS is a good micro!

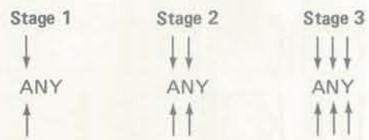
the first steps

Perusal by the author of books about programmed learning, or CAI in particular, revealed some sound approaches for presenting material, but it was always assumed that the student could read. Of course techniques with show cards, and the like, have been used as an aid to teaching word recognition but somehow nothing seemed to reach to the heart of Peter's problem. Basically a prescription for Peter's case would include:

(1) establishing left-to-rightness of words (on account of his dyslexia),
(2) rote learning of simple basic words,
(3) reading of stories containing extensive use of the simple basic words,
(4) rote learning of more involved basic words,
(5) and so on.

Unfortunately Peter was always stuck at step (2) of the prescription, and since someone has to laboriously write out the words, every time, for step (1), this step was ignored. If only someone, or something, was available with the necessary patience and persistence to follow the prescription. Now do you see that TRS was "just what the doctor ordered," for a micro is extremely patient and persistent.

The first program written to match the prescription for steps (1) and (2) consisted of display of words from the basic Dolch word list of 220 words that should be recognized on sight. Left to rightness was established by progressive generation of arrows from above and below the word. For example . . .



(Pardon the simplicity of this approach but you must remember the lack of string manipulation features—but hold on, see later for an improved method.) Peter had to say the word out loud before a continuously running cassette machine came forth with the same word. After this visual display-audio verification process finished for a word, Peter would mark himself right (1) or wrong (0), then at a "bip" from the cassette he would release the ENTER key. (This was the, simple way of keeping everything together.) After finishing 10 words a (graphics generated) rocket ship would reward him by moving up to a height depending on his results (also displayed). The method really worked, for a few days after starting him on the program he picked up a book, the first time ever of his own accord, and he slowly began to read. You see TRS has another element to "make the medicine go down"—the fun and novelty element.

We would have been stuck on step (2) after this experience except that timely help came from a University specialist educator. He piled on work for Peter in such great loads that a computer was needed even if simply on account of the sheer bulk of material. (I must ask him how parents not having access to a computer cope.)

Step (3) of Peter's basic prescription was satisfied with a beautiful story by Roger Farr, James Laffey and Carl Smith (from Taxonomy of Evaluation Techniques for Reading Programs) called "The Best Thing in the World." The story contains the 220 Dolch words imbedded in it. TRS was used to display lines of the story at a preselected rate of so many words a minute. Peter began with 20 words a minute and was in difficulty, but a few weeks later he had no trouble reading the story at 50 words a minute. (This result is good even allowing for the effect of retention.) Fortunately the story used has its own appeal and this has faithfully introduced Peter to the delights of reading.

the way

Over the past three months Peter has been introduced to more than 2,000 words and several stories using the TRS. Most of the words, however, have been taught using the technique described in this present section. When considering CAI with visual display and audio verification at least two different approaches appear depending on the way the audio is used. The two approaches are:

(1) the pacer technique—here the audio comes from an uninterrupted cassette recording, hence a fixed pace is maintained, *or*

(2) the timer technique—here the audio comes from segments of a computer activated cassette recording, hence the student determines the pace.

The difficulty with the first approach is that the pace needs to be set for the student about to begin learning and then the recording must be remade at a faster pace for revision work. Should the latter recording not be made frustrations can develop during revision work (after all the student is not likely to have as much patience with the machine as it originally had with him). The difficulty with the second approach is that of implementation on the TRS-80. The method calls for asynchronous interruption that follows the pace of the student through a word list and Level I BASIC apparently does not make available the necessary features. A method of overcoming this difficulty is presented here.

Level I BASIC has three graphic commands:
(1) SET (X,Y)—to turn on a graphics element positioned at X,Y on the screen,
(2) RESET (X,Y)—to turn off a graphics element at X,Y,
(3) POINT (X,Y)—to return 1 or 0 depending on whether an element is present at X,Y or not.

These commands, plus the fortunate connection of the keyboard to the screen even when the computer is otherwise engaged, enables asynchronous attention to be simulated. The idea is that the screen cursor, which can be addressed to any position on the screen, is placed near some graphics elements. Say the space bar is depressed; then a graphics element to the immediate right is destroyed (that is, overwritten with a blank) even though the computer is busy doing other things. The program can then make a regular check on the presence or absence of the element under discussion and hence change the course of the program appropriately. In the word timer program to follow, the author uses the four graphics elements which are destroyed by different key combinations to give four alternative courses of action following the signalling of "asynchronous attention."

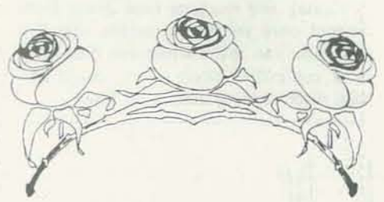
The features discussed, plus elementary logic operations such as:

D = (D>0) – (D<0), (D = the sign of D),

and the audio cassette activation statement:
PRINT #–(a four second write to cassette but with the cassette set to play mode instead of record), are used in the word timer program developed for most of Peter's word lists.

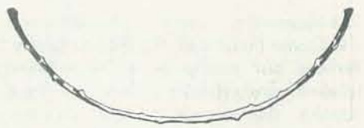
Other programs have been developed but it was thought best to present a fully documented version of one program in the hope that at least some other young girl or boy may be helped to read. The approach will not overcome all reading difficulties but for some there is hope that, like Peter, they may be born again into the wonderful world of words.

Dear micro owner, there is much need to help the handicapped. I hope that this article may, in some small way, arouse a new zeal for work in this area. Look for a problem that you can tackle on your own micro.



acknowledgements

In October (1977) an editorial in *COM3* by Timothy Mowchanuk of Essendon Grammar School, made an appeal for micros to be used to help the handicapped—this set the stage for the present work. From May, with the arrival of the TRS-80, Mary, Peter's mother, has made many sacrifices—for software development takes time, lots of time. This June and since, an enthusiastic educator, Glen Campbell, of Macquarie University has been supplying the author with material and diagnostic help for Peter. Without their insight and help no such testimony would have been possible—Peter can now read.



word timer program for trs-80

NOTE — lower case words are comments and are not part of the program.

```
110  REM WORD TIMER
112  CLS:P.''KEYS: SPACE = NEXT, ← = BACK, ENTER*1 = PAUSE, * = SEARCH''
115  H = 90:READP,N:M = -1:Y = 0
117  P.:P.''ENTER FOR VOICE CASSETTE'':P.'' N NOT WANTED''
118  P.'' M MACHINE SPEAKS FIRST'':P.'' Y YOU SPEAK FIRST''
119  IN.D:D = (D>0) - (D<0):C = (D>0)          d = sign(d) in case extraneous reply
120  F.Z = 1TON:Q = C                          loop over sections (pardon the next
                                               at 134)
130  IFZ = 1CLS:F.Y = 6TO47:S.(113,Y):S.(114,Y):   a wall to stop microbobs
     N.Y
131  B = 0:W = 1:A = 378:P.ATH,''''
132  P.AT0,''WORD GROUP'';P.'' SECT'';Z;'' . . . Y OR N TO PROCEED'';
133  Y = 0:IN.X:P.AT0,'''';:IFX<>YT.190
134  GOS.6500:IFM<0T.N.Z:P.AT0,'':END. RECORD TIMES'';:END
136  S.(3,2):S.(6,2):S.(1,3):S.(8,3)           set 'asynchronous' interruption
                                               elements
137  IFB>0T.160                                initial position of a microbot
140  X = 11:Y = 42-6*(Z-1)                     position of calculated 'sec/word'
145  A = 954-128*(Z-1)                         so far
147  P.AT896-128*(Z-1),Z;
150  P.AT186,''WORD'';:P.AT314,''SEC'';:IFZ<6P.AT442,''SEC/W'';
160  IFW<0P.AT2,'''';:G.171                    timer 'tick'
165  B = B+1
170  P.AT378,B;:P.AT2,'''';:GOS.9404           move microbot
171  IFP.(8,3) = 0T.250                        ,↑,'©'enter,enter' for search
172  IFP.(3,2) = 0T.200                        ,©,',←' for backspace
173  IFP.(1,3)<>0T.176                         ©',',''enter' for pause
174  IFW>0P.AT1,'''';:IN.B$:G.177              pause — 'break with enter
175  W = 4:Q = C:G.134                         restore to normal after search
176  IFP.(6,2) = P.(1,3)T.160                  ',●,' 'space or any other key' for
                                               advance
                                               voice if user first
177  E = 0:GOS.905:G.134
190  READL:F.I = 1TO8:READJ:N.I
195  F.I = 1TOL:READA$:N.I
196  M = -999:G.134
200  P.ATH,'''';:P.AT2,'''';
201  REST:M = M+2-(Q = 1)                      backspace one word
202  READJ,L:Q = Q-(Q>1)-(D<0)*(Q = 0)         Q denotes a variable
205  F.K = 1TOZ
210  READL:F.I = 1TO8:READJ:N.I
215  IFK = ZO = O-2+(O = 1):L = O
```

```
220  IFL<=0T.240
230  F.I = 1TOL:READA$:N.I
240  N.K:G.134
250  W = -4:Q = 999:G.134                      turn on search mode — break with
                                               enter
900  REM VOICE
905  IFQ<>0T.920                               voice response only if written words
                                               are in phase
910  IFE = DP.#                                relies on only the remote plug being
                                               connected
915  RET.                                      to give about 4 seconds of voice
                                               response
920  Q = Q+(E = 0):RET.
6500 REM FETCH
6501 IFM = -999M = -1:RET.
6502 IFM<0READM:F.L = 1TO8:READA(L):N.L:O = 0
6504 P.ATH,'''';P.AT250,O+(M<>0);:P.ATA,INT(10*B/(O+(O = 0))+.5)/10;

6505 M = M-1:IFM<0RET.
6506 READA$:O = O+1
6508 IFW<0P.ATH,A$;:RET.
6510 F.L = 1TO8
6512 P.ATH+7*L-32,''      '';:P.ATH+7*L-25,'' * = →'';
                                               rocket ship
6515 P.ATH,A$;
6516 IFL = 1E = -1:GOS.905                     voice if machine first
6517 IFL<8P.ATH+L,'''';                        partial word, eg.
                                               RE blanks to erase
6520 F.K = 1TOA(L):N.K                         delay
6525 N.L:RET.
9400 REM OK→?                                  microbot driver — an abridged
                                               version
9404 IFW>1T.9440
9406 GOS.9502                                  fine 'tuning' for 1 s delay —
9410 F.T = 1TO40:N.T                           adjust if necessary
9420 W = 4:RET.                                check that all clear to move
9440 IFP.(X+3,Y) = 1T.9450                     right
9442 G.9406                                    generate another microbot if not
9450 W = 1:X = 11:G.9406
9500 REM MICROBOT                              graphics generation — an abridged
                                               version
9502 IFW>1T.9542
9512 F.U = 46TO50:F.T = 127TO129:S.(X+T,Y+U):N.T:N.U
9514 R.(X+127,Y+47):R.(X+1,Y+47):R.(X,Y+2)
9516 S.(X+126,Y):S.(X+2,Y):RET.
9542 F.T = 126TO127:F.U = 46TO50:R.(X+T,Y+U):N.U:N.T:T = 1

9544 X = X+T:G.9512

     ——— then follows data for the actual words required to a maximum
                                               of 6 sections, e.g. ———

10000 REM GROUP & NO.,(NO IN SECT, DELAY EA LETTER(8),(WORD1,WORD2, . . .))
10001 D.17,3                                   arbitary group no for this
                                               collection of words, no of sections
      section 1
10100 D.42                                     number of words in section 1
10101 D.0,150,150,150,0,0,0,0                  delays (1/500s) after first
                                               8 letters
      words may be up to 16 characters in length
10102 D.REBUILD,REREAD,REVISE, . . .           required words
10104 D.RELATE,REFUND,REGARD, . . .
      section 2
10200 D.50                                     number of words in section 2
10201 D.120,120,120,120,0,0,0                  delays
10202 D.PEDDLER,RUBBER, . . .                  required words
      section 3 . . .
```

overview of the run

The program may be run in three different ways depending on the requirement with a voice response cassette. The three ways are . . .

N— no voice cassette to be used,
M— machine, that is the combination TRS-80 and cassette recorder, speaks first to enable the student to obtain a preview of the group of words, or
Y— you speak first and the machine verifies what you have said with its spoken reply—the normal mode when a voice cassette is available.

A supervisor would need to be present to verify responses when no voice cassette is available and in any case would normally be required to set up the machine at the start.

The aim of the program is to improve the student's ability and speed at recognizing words from prescribed word lists. Decoding skills are exercised even perhaps with the use of nonsense words. In any case similarly blended words should be collected together to form sections of a program, for example, words beginning with "re," etc. An accurate student's record of performance should be kept for each section of words. The program calculates an average "lethargy," seconds per word, for each section. This average lethargy would be recorded along with comments about certain word difficulties observed by a supervisor sitting through a session. After several attempts with the same section of words, but at later times, the lethargy should decrease. In the author's experience with Peter, and his word lists covering over 2,000 words, the lethargy usually starts about four or five s/word and after several attempts, days later, usually drops to about one or two s/word. This suspected improvement is verifiable with independent tests. (A lethargy of one s/word is a minimum since the timer only permits interruption every second.)

The program builds up a word from left to right (usually over a period of time such as a second) with a left to right moving rocket ship racing through the word as it is being built. Both techniques strongly emphasize the left to right manner of reading which is important for a dyslexic child. Also a "waltzing" microbot beats out the time and provides a pleasant reinforcement to speed of reading.

Establishing a voice response cassette (if required)

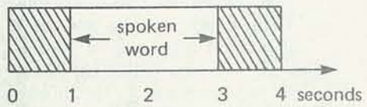
The following has been the practice of the author in setting up a voice cassette to go with the word timer program.

Step 1—Use a different cassette than that used to store the program as program updates may otherwise unintentionally destroy some of the recording. A C60 tape of reasonable quality has been found to be preferable.

Step 2—After a forward spacing of tape to move it clear of the non-recording surface, record a regular "count down" process to enable the starting position to be established accurately with the ear. (The playing of an octave of the C-scale is not a bad idea.)

Step 3—As in all uses of a voice response cassette, only the remote plug should be connected to the recorder and if possible a constant power source (the mains) should be used.

Set the cassette machine to the record position and simply run the program in the Y reply mode, "you speak first." After any key is depressed on the TRS-80 the cassette machine is activated for about four seconds. During this activation it is important that the word on display is said at about the middle of the time segment as illustrated . . .



After some practice, "centering the spoken word" is easily achieved. The reason for doing this is that the shaded areas in the illustration represent the uncertainty of tape position as it is being played back.

Step 4—At the start of each section it is a good practice to record a constant pitched sound for the first second, prior to the first spoken word. It is then possible to play through the cassette to detect the start of a new section.

Running the program

Assuming that a supervisor has set up a session, the steps required of a student are as follows.

Step 1—For each section of the group of words the student is asked the question:

WORD GROUP □ SECT □ . . . Y OR N TO PROCEED?

Normally he would reply Y (for yes) and ENTER from the keyboard. Should a section be required to be skipped for some reason the reply would be N (for no). In this case, care would be required with the positioning of a voice response cassette if one is being used.

Step 2—Assuming that the machine has not been prompted to speak first, the student watches the word being formed on the screen and then he decodes it and speaks the word. After saying the word he depresses any key, usually the space bar, (i) to interrupt the "waltzing" microbot, (ii) to activate the voice cassette, if one is available, to supply him with audio verification (or here the supervisor speaks) and (iii) to advance to the next word of the section. And so on for the remainder of the section.

Should a supervisor be available through the session the responses he can obtain from the machine are given below.

Response 1—To go back a word depress the "←" key. (If a voice cassette is being used the program will skip the voice until screen and cassette are again in phase.)

Response 2—To pause the timer, etc. in order to explain some detailed point depress the ENTER key. A further push of the same key will cause the program to move on to the next word.

Response 3—To skip backwards or forwards through the section of words, enter the search mode with two depressions of the ENTER key in quick succession. In the search mode everything "freezes" except the word display and here the word is simply flashed on the screen for speed of response instead of being built-up. (The mode is also useful for verification of a freshly typed-in section of words.) Exit from the mode is achieved by depressing the ENTER key (once).

To restore phase with a voice response cassette it is necessary to exit from the search mode at the word that was last spoken.

For an improperly made voice cassette, or some other unknown reason, the search approach may be necessary to bring visual and audio words together. In the author's experience it is sometimes necessary to do this. Indeed the voice "connection" is not as robust as the rest of the program and this is unfortunate. Even so the advantages of using a visual-audio program makes it worthwhile to attempt to overcome any "teething troubles."

Note—The "asynchronous" interruption procedure adopted in the program relies on the free response of the screen and its interaction with four sensing elements following depression of certain keys. However, for a small fraction of the time the cursor is away from "home" in order to increment the time. If a key is depressed then, it may be necessary to depress the key again following a lack of action of the required type.

A Letter from your Publisher

Changes, changes, changes!

The most evident change at People's Computer Company is that *People's Computers* is now *Recreational Computing*, a more accurate reflection of the content of the magazine.

Improved publication schedules have resulted in better circulation department and wholesale distribution services. The ultimate result is that all of you should now be receiving your favorite computer magazines on a timely basis. For those of you who may be distressed to find that your newsstand has your magazine before you receive it in the mail, it's because we send our magazines to the distributors via United Parcel Service and must use second class mail for our subscribers. Thanks for your understanding.

We are now accepting selected advertising. All publishers are having problems these days with shortages caused by the skyrocketing costs of materials, outside labor, shipping, etc. Our aim is to continue to supply you with a high quality product without corresponding increases in price.

I'd like to mention that we welcome and encourage retaining and sustaining subscribers. The benefit such a subscriber receives, other than the tax deduction, is the good feeling one gets from helping something succeed that one believes in. Rates are listed in the front section of this issue.

We have formulated a new policy to make our company eligible for foundation or corporation grants. Please contact me if you wish further information, or if you know of a foundation or corporation interested in contributing funds to one or more of the magazines published by our non-profit educational organization.

We value you as a reader and we will continue to furnish you with high-quality, innovative material. Your ideas and suggestions are always welcome. Please write or call me at People's Computer Company, P.O. Box E, Menlo Park, California 94025, telephone (415) 323-3111.

Sincerely,

Willard J. Holden

Willard Holden
Publisher

Round 2: BASIC

BY S. RAVN-JENSEN

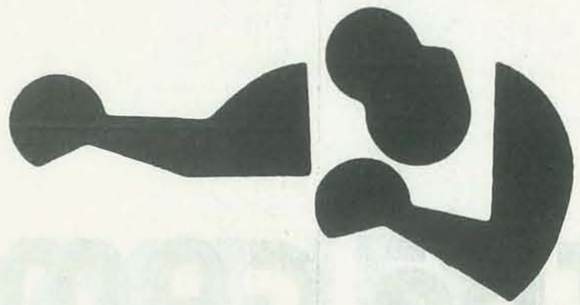
In the Jan-Feb 1978 issue of this magazine, David Mundie made a comparison of PASCAL & BASIC. He used a form of the Master Mind game as a program example. Part of David's conclusion was that PASCAL is inherently flexible while BASIC flexibility is a function of designing new versions of the entire language.

From Denmark comes a reply. S. Ravn-Jensen sends a version of Master Mind written in a powerful BASIC dialect. Why choose the poorest version of BASIC is the question Ravn-Jensen poses.

Well, only the readers can truly decide this debate. To help you in your labors, all three versions of the program appear below. –RZ.

```
LIST
0010 REM PROGRAM 'BASIC IS BETTER THAN THAT'
0020 REM DAVID A MUNDIE'S MASTER MIND PROGRAM
0030 REM REWRITTEN TO BASIC BY S. RAVN-JENSEN
0040 REM ENDRUPVEJ 115B, DK 2400 NV, DENMARK
0050 REM
0060 REM (W ITH THE CHANGES NEEDED AND WITHOUT
0070 REM OVERDIMENSIONING THE ARRAYS)
0080 REM 780914
0090 REM
0100 PROC NEWGAME
0110   DIM LO$(1),HI$(1)
0120   INPUT "LOW  CHARACTER ?  ",LO$
0130   INPUT "HIGH CHARACTER ?  ",HI$
0140   INPUT "NO.  OF  CHAR  ?  ",NUMCH
0150   REM NUELEM IS THE NUMBER OF ELEMENTS IN THE SET
0160   LET NUELEM=ORD(HI$)-ORD(LO$)
0170   REM MAXTR IS A BETTER LIMIT THAN MAXMAX
0180   LET MAXTR=NUMCH+NUELEM; EOG=0
0190   DIM TARG$(NUELEM),GUES$(NUELEM),CH$(1),OLDG$(MAXTR,NUELEM)
0200   DIM OLDB(MAXTR),OLDW(MAXTR),MATCH$(NUELEM)
0210   REM INITIALISATION OF TARG$ AND GUES$ NOT NEEDED
0220 ENDPROC  (NEWGAME)
0230 REM
0240 PROC NEWROUND
0250   LET EOR=0; TRY=0
0260   FOR I=1 TO NUMCH
0270     LET TARG$(I)=CHR(ORD(LO$)+INT(RND(0)*NUELEM))
0280   NEXT I
0290 ENDPROC  (NEWROUND)
0300 REM
0310 PROC COMMAND
0320   REM PROCEDURE 'TALLY' NOT NEEDED
0330   PRINT
0340   INPUT "   COMMAND ?  ",CH$
0350   CASE CH$ OF
0360     PRINT "ILLEGAL INPUT --SORRY"
0370     PRINT "THE COMMANDS ARE 'R', 'Q', 'S', OR 'C'"
0380     PRINT
0390   WHEN "R"
0400     FOR I=1 TO TRY
0410       PRINT OLDB(I);" B  ";OLDW(I);" W     ";OLDG$(I)
0420     NEXT I
0430   WHEN "Q"
0440     PRINT "ANSWER IS:   ";TARG$
0450     LET EOR=1
0460   WHEN "S"
0470     LET EOR=1; EOG=1
0480   WHEN "C"
0490     LET I=0
0500     PRINT "GUESS :  ";
0510     REPEAT
0520       LET I=I+1; BAD=0
0530       INPUT "",GUES$(I);
0540       IF GUES$(I)<LO$ OR GUES$(I)>HI$ THEN LET BAD=1
0550     UNTIL I=NUMCH OR BAD
0560     PRINT "   ";
0570     IF BAD THEN
0580       PRINT GUES$(I);" IS A BAD CHARACTER"
0590     ELSE
0600       IF GUES$=TARG$ THEN
0610         PRINT "YOU GUESSED IT !!!"
0620         LET EOR=1
0630       ELSE
0640         IF TRY=MAXTR THEN
0650           PRINT "YOU ARE LOST"
0660           PRINT "ANSWER IS :  ";TARG$
0670           LET EOR=1
0680         ELSE
0690           LET BLA=0; WHI=0; TRY=TRY+1
0700           FOR I=1 TO NUMCH
0710             LET MATCH$(I)="F"
0720             IF GUES$(I)=TARG$(I) THEN LET MATCH$(I)="T"; BLA=BLA+
0730             REM (COUNT BLACKS)
0740           NEXT I
0750           FOR I=1 TO NUMCH
0760             IF GUES$(I)<>TARG$(I) THEN
0770               LET J=0
0780               REPEAT
0790                 LET J=J+1; WM=0
0800                 IF GUES$(I)=TARG$(J) AND MATCH$(J)="F" THEN LET WM=1
0810                 IF WM THEN LET MATCH$(J)="T"; WHI=WHI+1
0820               UNTIL WM OR J=NUMCH
0830             ENDIF (COUNT WHITES)
0840           NEXT I
0850           PRINT "  B : ";BLA;"   W : ";WHI
0860           LET OLDG$(TRY)=GUES$; OLDB(TRY)=BLA; OLDW(TRY)=WHI
0870         ENDIF
0880       ENDIF
0890     ENDIF
0900   ENDCASE  ( 'R', 'Q', 'S'  OR  'C' )
0910 ENDPROC  (COMMAND)
0920 REM
0930 REM MAIN PROGRAM
0940 EXEC NEWGAME
0950 REPEAT ROUNDS
0960   EXEC NEWROUND
0970   REPEAT ACCEPTING COMMANDS
0980     EXEC COMMAND
0990   UNTIL EOR
1000 UNTIL EOG
1010 END OF MAIN
```

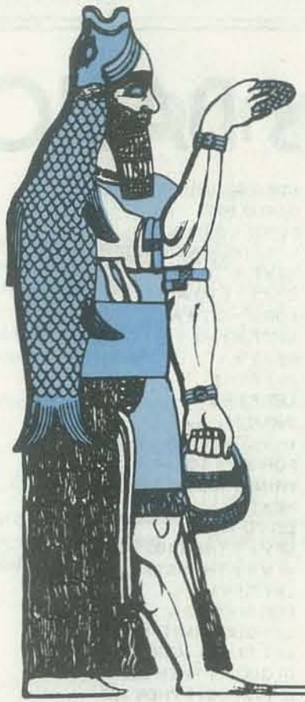

vs PASCAL



```
program banbasic(input,output);
const maxnumch=10; maxmax=71;
type token=packed array[1..maxnumch]of char;
var target,guess:token; hi,lo,ch:char; oldg:array 1..maxmax of token;
    i,j,try,maxtries,black,white,numchar:integer;
    oldb,oldw:packed array[1..maxmax]of integer;
    endofround,endofgame,wmatch,bad:boolean;
    matched:array[1..maxnumch]of boolean;
procedure newgame;
 begin endofgame:=false;
  for i:=1 to maxnumch do target[i]:=' '; guess:=target;
  writeln(' low character?'); readln(lo);
  writeln(' high character?'); readln(hi);
  writeln(' no. of characters?'); readln(numchar);
  maxtries:=numchar+ord(hi)-ord(lo)
 end;
procedure newround;
 begin endofround:=false; try:=0; for i:=1 to numchar do
  target[i]:=chr(ord(lo)+trunc(random(1)*(ord(hi)-ord(lo))))
 end;
procedure command;
  procedure tally(var i,color:integer);
   begin matched[i]:=true; color:=color+1 end;
 begin writeln(' command?'); readln(ch); case ch of
 'r':for i:=1 to try do writeln(oldb[i],'b',oldw[i],'w',oldg[i]);
 'q':begin writeln(' answer is: ',target); endofround:=true end;
 's': begin endofround:=true; endofgame:=true end;
 'c': begin i:=0; repeat i:=i+1; read(guess[i]);
       bad:=not(guess[i]in[lo..hi]) until (i=numchar)or(bad); readln;
       if bad then writeln(' bad character') else if guess=target then
       begin writeln(' you guessed it!'); endofround:=true
       end else if try=maxtries then
       begin writeln(' you are lost; answer is: ',target);endofround:=true
       end else
       begin black:=0; white:=0; try:=try+1;
        for i:=1 to numchar do matched[i]:=false;
        for i:=1 to numchar do if guess[i]=target[i]then tally(i,black);
        for i:=1 to numchar do if guess[i]<>target[i]then
        begin j:=0; repeat j:=j+1;
         wmatch:= (guess[i]=target[j]) and (not(matched[j]));
         if wmatch then tally(j,white) until (wmatch)or(j=numchar)
        end; writeln(' b',black,' w',white);
        oldg[try]:=guess; oldb[try]:=black; oldw[try]:=white
       end
      end
  end
 end;
begin newgame;
 repeat newround;
  repeat command until endofround
 until endofgame
end.
```

vs BASIC

```
10   DIM F(9),G(9),T(9),H(18,3)
20   GOSUB 560
30   FOR X = 0 TO A
40   LET T(X) = INT(RND(R)*B)+1
50   NEXT X
60   FOR I = 1 TO A+B+1
70   FOR X = 0 TO A
80   LET F(X) = 0
90   NEXT X
100  LET F1 = 0
110  LET F2 = 0
120  INPUT V
130  IF V<>0 THEN 180
140  FOR X = 1 TO I-1
150  PRINT H(X,0); "," ;H(X,1); "=" ;H(X,2)
160  NEXT X
170  GO TO 120
180  IF V = 1 THEN 480
190  IF V = 2 THEN 670
200  LET T1 = V
210  FOR X = 0 TO A
220  LET G(X) = INT(T1/(10**(A-X)))
230  LET T1 = T1-G(X)*(10**(A-X))
240  IF G(X)<1 THEN 260
250  IF G(X)<B+1 THEN 280
260  PRINT "BAD NUMBER IN"; V
270  GO TO 70
280  IF G(X)<>T(X) THEN 310
290  LET F(X) = 1
300  LET F1 = F1+1
310  NEXT X
320  IF F1 = A+1 THEN 540
330  FOR Y = 0 TO A
340  IF T(Y) = G(Y) THEN 420
350  FOR X = 0 TO A
360  IF G(Y)<>T(X) THEN 410
370  IF F(X) = 1 THEN 410
380  LET F(X) = 1
390  LET F2 = F2+1
400  GO TO 420
410  NEXT X
420  NEXT Y
430  PRINT F1;",";F2
440  LET H(I,0) = F1
450  LET H(I,1) = F2
460  LET H(I,2) = V
470  NEXT I
480  LET V= 0
490  FOR X = 0 TO A
500  LET V = V+T(X)*(10**(A-X))
510  NEXT X
520  PRINT "ANSWER IS";V
530  GO TO 30
540  PRINT "YOU GUESSED IT"
550  GO TO 30
560  PRINT
570  PRINT "   DIGITS & MAX VALUE"
580  INPUT A,B
590  LET A = A-1
600  RETURN
610  END
```



The GAME of LIFE

BY HOWARD A. PEELLE

If Howard Peelle's PHANTNUM intrigued you in the Nov.-Dec. 1978 issue, you'll want to try his GAME OF LIFE, too. This APL version of that golden oldie is excerpted from his forthcoming book, Instructional Applications of Computers Using A Programming Language.—*LB*

INTRODUCTION

Beware! The "Game of Life" is delightful, intriguing and somewhat addictive. It is a non-competitive activity, played with simple rules—yet rules whose outcomes are deceptively difficult to predict. Actually, Life is not a game in the usual sense, in that there are no 'opponents' nor 'strategies' for winning. Instead, it is a simulation of a cellular automaton which involves you first as a creator, then as an observer of potentially endless series of symbolic patterns which seem to have a life of their own.

In its short history, Life has gained extraordinary popularity—especially among computer buffs. Invented by Cambridge University mathematician John H. Conway, Life was first publicized in this country by Martin Gardner in his Mathematical Games department of *Scientific American* magazine (October, 1970). Its ensuing popularity is due, in large part, to the computer. The accuracy and speed of the computer are indispensable for repeatedly executing the 'genetic laws' of Life. Indeed, much illicit computer time has been expended in generating displays of Life configurations—pulsating away on cathode ray tubes in various computer centers.

THE "GAME OF LIFE"

Life is played on a rectangular grid, where a 'colony' of 'organisms' is placed. An 'organism' is represented by a single symbol, such as *. Accordingly, a 'colony' of organisms is a collection of these symbols arrayed in some pattern. The player can place symbols on the grid in a configuration of his/her choosing, or conceivably one could place the symbols at random.

The game begins when a colony has been specified and certain rules are applied. (See the rules below). The colony is then transformed into a new colony—the next 'generation'—by way of these rules. This process is repeated, possibly indefinitely.

RULES

Successive generations of a colony are reproduced according to the following two "laws of Life":

1. *The Law of Survival*
   Each organism with 2 or 3 neighbors *survives* to the next generation.

   For example,

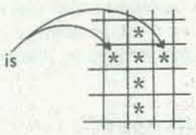
   In other words:

   Each organism with 4 or more neighbors dies from 'overcrowding'.

   Each organism with 1 or fewer neighbors dies from 'isolation'.

2. *The Law of Birth*
   Each empty space with exactly 3 neighbors has a *birth* of a new organism in the next generation.

   For example,

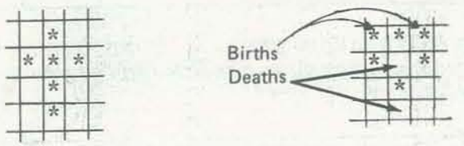
   

   Births occur where there is optimal 'nurture' in the neighborhood.

Note that a "neighbor" is defined as an organism present in any adjoining space—horizontally or vertically or diagonally. There are a total of eight such adjoining spaces (marked by . below):

The "laws of Life" are applied simultaneously, so that one generation of a colony gives way to the next generation. For example,

Generation 1: becomes Generation 2:



OUTCOMES

The Game of Life goes on—generation by generation—until one of the following outcomes occurs: (1) All of the organisms die (in which case the game terminates); (2) The colony reaches a stable configuration (either when the pattern of organisms in one generation produces the same pattern in the next generation or when any one of the previous patterns is repeated—producing periodic "cycling" of generations); or (3) The colony grows indefinitely.

Examples of each of these outcomes are illustrated later.

OBJECTIVES

Although the learning objectives of this game are unlike most gaming activities (they are more akin to those for simulations), the educational benefits are undeniably rich. There is little payoff in outguessing the computer or pursuing optimal strategies, but the student might:

Draw Analogies with Other Systems, e.g.:
- ecology (population dynamics)
- economics (resource allocation)
- biology (genetics)
- chemistry (molecular interaction)
- cybernetics (information processing)
- mathematics (differential equations)
- biomedicine (cell and organ simulation)
- physics (magnetic fields)

Engage Related Mathematical Questions, e.g.:

- Does there exist a colony which will grow forever (never dying out and never repeating itself)?[1]
- Are there any algorithms for "backward synthesis"—i.e., determining predecessor ("parent") colonies?
- Do there exist "original" colonies which may never be produced by the "laws of Life," i.e., have *no* parents? (This is known as the "Garden of Eden" question.)[2]
- Is there a colony which has a parent but no grandparent?

Examine the Aesthetics:

- Seeing the Game of Life as an art form, one could study individual patterns and their dynamics for aesthetic qualities.

Think of Related Games or Simulations:

- (See Extensions of Life further on in this article.)

[1] This question was first raised by John H. Conway, who offered $50 prize for its answer. It has since been answered. (See the "glider" colony on page 19 for a clue; or write Robert T. Wainwright, 1280 Eden's Road, Yorktown Heights, NY, 10598 for back issues of "Life-line," a newsletter for enthusiasts.)

[2] Banks and Ward, at M.I.T., have shown that a Garden-of-Eden pattern—an "orphan"—is contained in a 9 by 33 grid. Can you find it?

PROGRAMMING THE GAME OF LIFE ON THE COMPUTER

Although Life is certainly rewarding when performed by hand, it takes on added dimensions when programmed on a computer. The computer can, of course, facilitate setting up the game—either placing organisms in a colony at the user's discretion or placing them randomly. But, more important, the computer then causes the various patterns to unfold much more rapidly and accurately than one could attempt manually. When the necessary calculations are performed and Life is viewed on a graphic display terminal, the human player is witness to a remarkable kaleidoscopic show.

The Game of Life is described in the following programs, written in APL.

First the main program:

```
     ∇GAMEOFLIFE
[1] 'WELCOME TO THE GAME OF LIFE.'
[2] 'DO YOU KNOW THE RULES?'
[3] →READY IF ∧/'YES' ∈⍞
[4] LAWSOFLIFE
[5] READY:'ENTER YOUR COLONY NOW.'
[6] COLONY←ENTER
[7] 'HERE IS YOUR LIFE PORTRAIT:'
[8] LIFE COLONY
    ∇
```

This program welcomes the player, explains the rules (if requested), uses a sub-program to ENTER a COLONY, and finally calls a sub-program to display the LIFE portrait of the COLONY.

The name of this program is GAMEOFLIFE. (Program names are underlined here for easy identification.) The del symbols (∇) mark the beginning and the end of the program.

Line [1] and [2] print out text. Line [3] accepts the user's answer (⍞) and branches to a line called READY if all of (∧/) the letters 'YES' are found in (∈) the answer. If not, it goes on to the next line [4] which executes sub-program LAWSOFLIFE. Line [5], READY, prints text. Line [6] utilizes sub-program ENTER to specify a COLONY. Line [7] prints text. Line [8] executes sub-program LIFE for the particular COLONY.

IF is a sub-program used to make branching commands easily readable. Its definition is:

```
    ∇BRANCH←LINE IF CONDITION
[1] BRANCH←CONDITION/LINE
    ∇
```

The rules are embodied in a sub-program called LAWSOFLIFE:

```
     ∇LAWSOFLIFE
[1]  ''
[2]  'THE GAME OF LIFE BEGINS WITH A COLONY OF ORGANISMS.'
[3]  'FOR EXAMPLE, YOU COULD BEGIN WITH A COLONY LIKE THIS:'
[4]  ''
[5]  '    * '
[6]  '  ***'
[7]  '    * '
[8]  '    * '
[9]  ''
[10] 'WHERE EACH  *  REPRESENTS AN ORGANISM.'
[11] ''
[12] 'THEN SUCCESSIVE GENERATIONS OF THE COLONY ARE REPRODUCED'
[13] 'ACCORDING TO THE FOLLOWING ''LAWS OF LIFE'':'
[14] ''
[15] '   1. EACH ORGANISM WITH 2  OR 3  NEIGHBORS'
[16] '      SURVIVES TO THE SUCCEEDING GENERATION.'
[17] ''
[18] '   2. EACH BLANK SPACE WITH  3  NEIGHBORS WILL HAVE A BIRTH'
[19] '      OF A NEW ORGANISM IN THE SUCCEEDING GENERATION.'
[20] ''
     ∇
```

Sub-program ENTER is a mechanism which facilitates entering symbols into a matrix (one row at a time) and may be omitted if the player specifies the COLONY by himself. (See Appendix for the definition of ENTER.) Note, though, that the result of ENTER is a COLONY of 0s and 1s—where the 0s represent blank spaces and the 1s represent organisms.

```
     ∇ LIFE COLONY ; GENERATION
[1]    GENERATION←0
[2]   NEXT:GENERATION←GENERATION+1
[3]    PRINT COLONY
[4]    COLONY←EVOLVE COLONY
[5]   →END IF 0=+/+/COLONY
[6]   →NEXT
[7]   END:      'LIFE HAS EXPIRED AFTER ';GENERATION;'  GENERATIONS.'
     ∇
```

Sub-program LIFE (shown above) is the structure which performs the basic simulation procedures: It begins at GENERATION 0; it specifies the NEXT GENERATION to be one greater than the previous GENERATION; it PRINTs the COLONY (see sub-program PRINT below); it EVOLVEs a given COLONY to become a new COLONY (see sub-program EVOLVE below); it goes to the END of the program IF the sum of the organisms in the COLONY is ever equal to 0 (meaning there are no organisms left); otherwise it repeats the process with the NEXT generation.

```
     ∇ PRINT COLONY
[1]    ''
[2]    GENERATION
[3]    ' * '[COLONY+1]
     ∇
```

This program PRINTs a blank line (for spacing), then the current GENERATION number, followed by a portrait of the COLONY — using ' ' (blanks) and * (asterisks) in place of 0s and 1s, respectively.

Program EVOLVE is at the heart of this simulation model.

```
    ∇ NEW←EVOLVE COLONY
[1]   COLONY←0 BORDER COLONY
[2]   NEIGHBORS←SURVEY COLONY
[3]   BIRTHS←(~COLONY)∧(NEIGHBORS=3)
[4]   SURVIVORS←COLONY∧(NEIGHBORS=2)∨(NEIGHBORS=3)
[5]   NEW←BIRTHS∨SURVIVORS
    ∇
```

The program takes as input an existing COLONY (represented in 0s and 1s) and produces a NEW colony. It begins, on line [1], by placing a BORDER of 0s (representing blank spaces) around the COLONY (to allow room for possible births).

```
    ∇ BORDERED←SYMBOL BORDER MATRIX
[1]   BORDERED←SYMBOL,
        (SYMBOL, MATRIX,      SYMBOL)
                 ,SYMBOL
    ∇
```

Program BORDER accomplishes this by appending the SYMBOL (0) to the top, bottom, left, and right of the colony MATRIX.

Then, on line [2], a SURVEY is conducted to determine the number of NEIGHBORS associated with each organism in the COLONY.

```
    ∇ NEIGHBORS←SURVEY COLONY
[1]   NEIGHBORS←(1⌽COLONY)+(¯1⌽COLONY)
                +(1⊖COLONY)+(¯1⊖COLONY)
            +(¯1⊖1⌽COLONY)+(1⊖1⌽COLONY)
          +(¯1⊖¯1⌽COLONY)+(1⊖¯1⌽COLONY)
    ∇
```

This is performed by simultaneous array calculations in program SURVEY. (Details are not discussed here.) Suffice it to say that the result NEIGHBORS is a matrix identical in size to COLONY containing the number of NEIGHBORS for each element of COLONY.

Then, on lines [3] and [4], the BIRTHS and SURVIVORS are computed. First, all BIRTHS are found simultaneously. They are found precisely where COLONY has blank spaces (~ COLONY produces 1s where there are 0s in COLONY) *and* where matrix NEIGHBORS is equal to 3. (∧ is the and function in APL.)

All SURVIVORS are found similarly, except that they are located where there are organisms in COLONY (1s) *and* where there are NEIGHBORS which equal 2 *or* 3. (∨ is the or function in APL.)

Finally, on line [5], the NEW colony is a matrix where there are 1s in BIRTHS *or* SURVIVORS.

PLAYING THE GAME OF LIFE VIA COMPUTER

One is now ready to play the Game of Life, with the assistance of the computer. Starting at the beginning, type GAMEOFLIFE.


```
      GAMEOFLIFE

WELCOME TO THE GAME OF LIFE.
DO YOU KNOW THE RULES?
NO

THE GAME OF LIFE BEGINS WITH A COLONY OF ORGANISMS.
FOR EXAMPLE. YOU COULD BEGIN WITH A COLONY LIKE THIS:

     *
    ***
     *
     *

WHERE EACH *  REPRESENTS AN ORGANISM.

THEN SUCCESSIVE GENERATIONS OF THE COLONY ARE REPRODUCED
ACCORDING TO THE FOLLOWING 'LAWS OF LIFE':

  1. EACH ORGANISM WITH  2  OR  3  NEIGHBORS
     SURVIVES TO THE SUCCEEDING GENERATION.

  2. EACH BLANK SPACE WITH  3  NEIGHBORS WILL HAVE A BIRTH
     OF A NEW ORGANISM IN THE SUCCEEDING GENERATION.

ENTER YOUR COLONY NOW.

     *
    ***
     *
     *
```

The above configuration—called the Latin Cross—dies out in five generations, as shown on the next page.

HERE IS YOUR LIFE PORTRAIT:

```
1
 *
***
 *
 *

2
***
* *

3
  *
 * *
 * *

4
   *
  * *

5
  *
  *
```

LIFE HAS EXPIRED
AFTER 5 GENERATIONS.

Sample Life histories of some other colonies follow. (For purposes of illustration, assume that the colonies have been specified beforehand, rather than going through the procedure just shown.)

LIFE HOOK

```
1
**
*
*

2
 **
**

3
***
***

4
 *
* *
* *
 *

5
 *
* *
* *
 *

6
 *
* *
* *
 *
```

Note that sub-program LIFE may be used directly if the colony is already established.

The outcome of this LIFE is a stable colony (a "still life").

At this point, the computer print-out was interrupted.

Some of the common "still life" forms include:

Beehive

```
 *
* *
* *
 *
```

Pond

```
 **
*  *
*  *
 **
```

Tub

```
 *
* *
 *
```

Block

```
**
**
```

Snake

```
* **
** *
```

Loaf

```
 **
*  *
* *
 *
```

Boat

```
 *
* *
 **
```

Barge

```
  *
* *
 * *
  *
```

Ship

```
 **
* *
 **
```

Long Boat

```
  *
* *
 * *
  **
```

Long Barge

```
   *
 * *
  * *
   *
```

Long Ship

```
  **
 * *
* *
 **
```

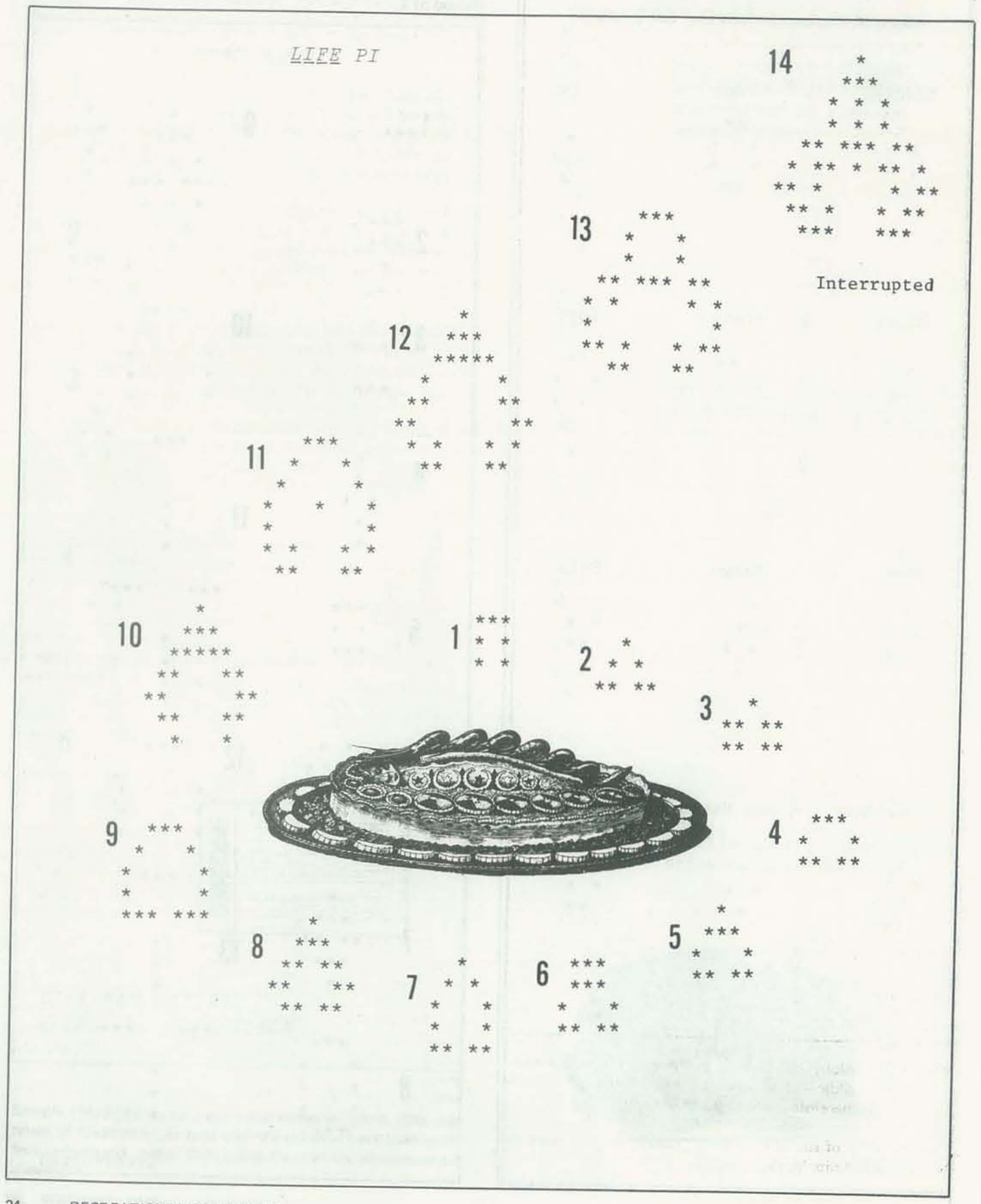
This initial colony eventually EVOLVEs into a cyclic pattern of "traffic lights," which after generation 9 flip-flop with a period of 2.

LIFE TACK

```
1
 *
***

2
***
***
 *

3
  *
 * *
***

4
  *
  *
 * *
  *
  *

5
 ***
 * *
 ***

6
  *
 * *
*  *
 * *
  *

7
 ***
** **
 ***
  *

8
 ***
*   *
*   *
*   *
 ***

9
   *
 ***
* * *
*** ***
* * *
 ***
   *

10
 ***
*   *
*   *
*   *

 ***

11
   *
   *
   *
***   ***

   *
   *
   *

12
 ***
*   *
*   *
*   *

 ***

13
   *
   *
   *
***   ***

   *
   *
   *
```

Interrupted

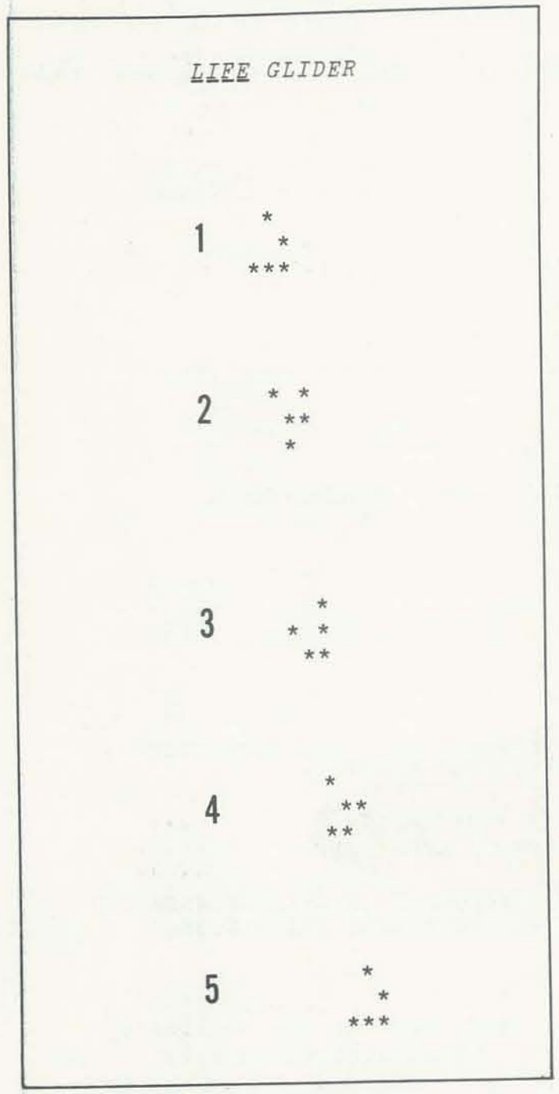
This colony appears to grow and grow, transcendentally.



Some colonies tend to move across the grid.

For example, the "glider"moves down and to the right one square every four generations before replicating itself.

(It is said to move at ¼ the speed of light.)



Note that a colony with a stable component which repeatedly sends off a glider—or any moving, self-sustaining "scouting party"—will therefore continue to grow forever.

The existence of such a colony—a "glider gun"—answers the question of infinite life raised earlier.

EXTENSIONS OF LIFE

Extensions of the Game of Life easily suggest themselves to the acclimated player. Some extra features which could be built into the programs developed here include:

1. *Changeable Laws of Life*
   Making the rules for survival and birth changeable allows the player to experiment with different Life "systems," e.g. changing the number of neighbors required for births to 2 generally yields more prolific communities.

2. *Variable Life Portraits*
   Allowing one to specify the interval of Life portraits becomes desirable when players wish to view colonies with extended longevity. E.g. a print-out once every 100 generations.

   The symbol used for displaying organisms can be changed at the player's discretion—say, from * to □. In addition, a different symbol, e.g. * could be used to identify the new organisms (births) in a colony.

3. *Display Orientation and Economy*
   Since a border of spaces is appended to the colony each generation, its position changes. The colony moves down and to the right (relative to the top left of the page) one space every generation.

   It is sometimes desirable to trim the extra rows and columns of spaces (on the borders only), especially when viewing the Life portraits of colonies which grow to large proportions.

   This technique is desirable since computer input-output devices are always constrained for practicality—about a hundred characters per horizontal line on a typical telecommunications terminal.

   This technique, however, has certain sacrifices. E.g., the true position of a colony, as it evolves, is lost. Only distances relative to organisms within the colony are preserved.

4. *Maxima*
   Specifying maxima in the game—either a maximum number of reproductive cycles, say 1000 generations, or a maximum size allowable for any colony, say 30 by 50—bounds the time invested in any one colony (the computer's time too!).

Of course, there are also variants of the game itself. For example, Life can be played (1) on non-rectangular grids, such as a hexagonal grid; (2) in three dimensions (or more?); (3) on cylindrical, toroidal (doughnut) or any number of other surfaces; (4) with boundaries, so that organisms reaching the edge either (a) fall off and die, (b) come up the other side—say after a latency period of one generation, or (c) reflect back at the same angle of approach; (5) with interacting species, e.g. "viruses;" and (6) with competing species. The possibilities for creative "Lifing" seem endless. Have fun! But beware . . . you may not be able to quit.

APPENDIX

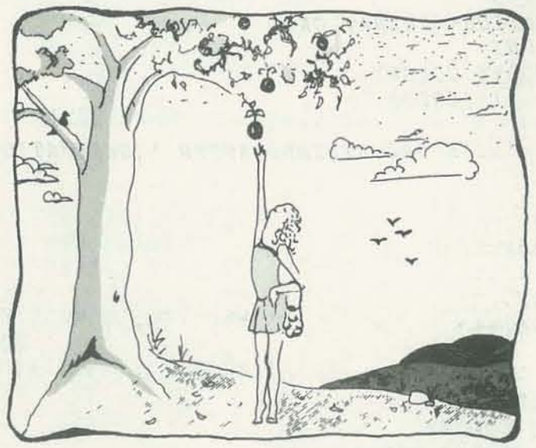
Complete Program Displays:

```
      )LOAD LIFE2
SAVED  06/14/74
      )FNS
BORDER  ENTER  EVOLVE  GAMEOFLIFE  IF  LAWSOFLIFE  LIFE  PRINT  SURVEY  TRIM

      ∇GAMEOFLIFE[□]∇
    ∇ GAMEOFLIFE
[1]     'WELCOME TO THE GAME OF LIFE.'
[2]     'DO YOU KNOW THE RULES?'
[3]     →READY IF ∧/'YES'∈⍞
[4]     LAWSOFLIFE
[5]    READY:'ENTER YOUR COLONY NOW.'
[6]     COLONY←ENTER
[7]     'HERE IS YOUR LIFE PORTRAIT:'
[8]     LIFE COLONY
    ∇

      ∇LAWSOFLIFE[□]∇
    ∇ LAWSOFLIFE
[1]     ''
[2]     'THE GAME OF LIFE BEGINS WITH A COLONY OF ORGANISMS.'
[3]     'FOR EXAMPLE, YOU COULD BEGIN WITH A COLONY LIKE THIS:'
[4]     ''
[5]     '       * '
[6]     '      ***'
[7]     '       * '
[8]     '       * '
[9]     ''
[10]    'WHERE EACH * REPRESENTS AN ORGANISM.'
[11]    ''
[12]    'THEN SUCCESSIVE GENERATIONS OF THE COLONY ARE REPRODUCED'
[13]    'ACCORDING TO THE FOLLOWING ''LAWS OF LIFE'':'
[14]    ''
[15]    '    1. EACH ORGANISM WITH 2 OR 3 NEIGHBORS'
[16]    '       SURVIVES TO THE SUCCEEDING GENRATION.'
[17]    ''
[18]    '    2. EACH BLANK SPACE WITH 3 NEIGHBORS WILL HAVE A BIRTH'
[19]    '       OF A NEW ORGANISM IN THE SUCCEEDING GENERATION.'
[20]    ''
    ∇

      ∇ENTER[□]∇
    ∇ MATRIX←ENTER;LINE
[1]     MATRIX←0 100⍴0
[2]     LINE←,⍞
[3]     →6 IF 0=⍴LINE
[4]     MATRIX←MATRIX,[1] ' ' ≠100↑LINE
[5]     →2
[6]     MATRIX←TRIM MATRIX
    ∇
```



```
      ∇LIFE[□]∇
    ∇ LIFE COLONY;GENERATION
[1]     GENERATION←0
[2]    NEXT:GENERATION←GENERATION+1
[3]     PRINT COLONY
[4]     COLONY←EVOLVE COLONY
[5]     →END IF 0=+/,/COLONY
[6]     →NEXT
[7]    END:       'LIFE HAS EXPIRED AFTER ';GENERATION;'  GENERATIONS.'
    ∇

      ∇PRINT[□]∇
    ∇ PRINT COLONY
[1]     ''
[2]     GENERATION
[3]     ' *'[COLONY+1]
    ∇

      ∇EVOLVE[□]∇
    ∇ NEW←EVOLVE COLONY
[1]     COLONY←0 BORDER COLONY
[2]     NEIGHBORS←SURVEY COLONY
[3]     BIRTHS←(~COLONY)∧NEIGHBORS=3
[4]     SURVIVORS←COLONY∧∨/NEIGHBORS∘.=2 3
[5]     NEW←BIRTHS∨SURVIVORS
    ∇

      ∇BORDER[□]∇
    ∇ BORDERED←SYMBOL BORDER MATRIX
[1]     BORDERED←SYMBOL,(SYMBOL,[1] MATRIX,[1] SYMBOL),SYMBOL
    ∇

      ∇SURVEY[□]∇
    ∇ Z←SURVEY X
[1]     Z←(1⌽X)+(¯1⌽X)+(1 ⊖ X)+(¯1 ⊖ X)+(¯1 ⊖ 1⌽X)+(1 ⊖ 1⌽X)+
                                          (¯1 ⊖ ¯1⌽X)+1 ⊖ ¯1⌽X
    ∇

      ∇TRIM[□]∇
    ∇ DOWN←TRIM MATRIX
[1]     DOWN←(0,¯1+(∨⌿ MATRIX)⍳1)↓(0,1-(∨⌿ ⌽MATRIX)⍳1)↓MATRIX
[2]     DOWN←(⌽0, ¯1+(∨/DOWN)⍳1)↓(⌽0,1-(∨/ ⊖ DOWN)⍳1)↓DOWN
    ∇

      ∇IF[□]∇
    ∇ BRANCH←LINE IF CONDITION
[1]     BRANCH←CONDITION/LINE
    ∇
```

PILOT FOR THE APPLE II



an extended Micro-PILOT interpreter

BY CHUCK CARPENTER

From our crack Apple II correspondent, Chuck Carpenter, comes this article on Micro-PILOT—the first, he says, of an everything-you've-always-wanted-to-know series. Here Chuck describes the interpreter written by fellow Texan Arley Dealey.

Arley is a student at Southern Methodist University, where he is majoring in underwater archaeology and doing programming on the side. To handle this sideline, Arley has formed a company, Magicke Software. Among his current projects—in addition to the PILOT interpreter described below—are a CO-PILOT text editor, an advanced text editor, and a disk utility package. He also writes programs for several small businesses in the area. As Chuck Carpenter puts it, "Arley has a talent for programming."

—LB

PILOT interpreters written in BASIC and machine languages are becoming increasingly available. The PILOT interpreter to be described here was written in Applesoft floating point BASIC by Arley Dealey. Apple II Micro-PILOT was first conceived early in 1978 from the program written by Chuck Shapiro, whose article on the subject appeared in the Sept-Oct 1977 issue of *People's Computers*. Arley's Micro-PILOT uses the same logic, but has several significant additions and extensions.

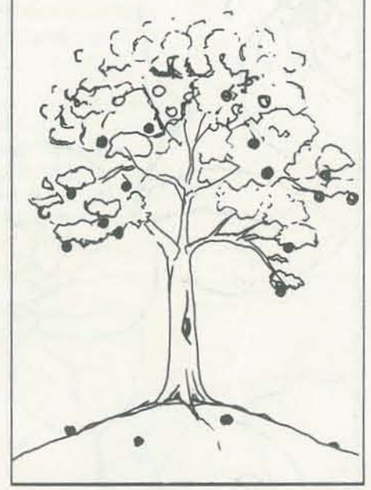
Here are some of the interpreter's features:

- All the commands and instructions used by C. Shapiro
- COMPUTE instruction added
- INVERSE and FLASH of characters or words
- Disk SAVE, LOAD and REPLACE commands
- Named programs
- Soft entry after exiting interpreter with BYE
- Paddle #1 (PDL 0) controls list speed
- Syntax error message
- Use of @ to inhibit carriage return
- Call peripherals from REQUEST using PR# (X)
- Suspend listing with space bar
- Continue listing with any key
- Line length limit warning bell

See page 31 for a complete list of Apple II Micro-PILOT features.

Listing of a program loaded from the disk

```
REQUEST? DLOAD
PROGRAM NAME?COMPUTE

REQUEST? LIST

      COMPUTE


 0  T:THIS IS A COMPUTE TEST
 1  T:
 2  C:2+3
 3  T:
 4  C:2/3
 5  T:
 6  C:2+3
 7  T:
 8  C:2-3
 9  T:
10  C:C+1
11  T:
12  C:C+1
13  T:
14  C:C+1
15  T:
16  C:C-1
17  T:
18  END:

REQUEST? RUN

RUN OF COMPUTE

THIS IS A COMPUTE TEST

2+3 = 6

2/3 = .666666667

2+3 = 5

2-3 = -1

C+1 = 1

C+1 = 2

C+1 = 3

C-1 = 2

END OF RUN
```



Example 2

Two examples showing the compute feature

Write a new program named COUNT

```
REQUEST? NEW
PROGRAM NAME?COUNT

 0?*BEGIN
 1?T:
 2?T: THIS PROGRAM COUNTS
 3?T:
 4?C:C+1
 5?J:*BEGIN
 6?END:
 7?DONE

REQUEST? LIST

      COUNT


 0 *BEGIN
 1  T:
 2  T: THIS PROGRAM COUNTS
 3  T:
 4  C:C+1
 5  J:*BEGIN
 6  END:

REQUEST? RUN

RUN OF COUNT


 THIS PROGRAM COUNTS

C+1 = 3

 THIS PROGRAM COUNTS

C+1 = 4

 THIS PROGRAM COUNTS

C+1 = 5

 THIS PROGRAM COUNTS

C+1 = 6
```

Loop stopped with Control-C

Apple II Micro-PILOT's inner workings are very similar to most current versions. If you have been reading about PILOT in past issues of *RC/PC*, you have an understanding of the language. (If not, the back issues are a good place to get up to speed.) There are some differences, however, and these will be discussed in more detail.

```
15  T: HOW DO YOU FEEL NOW $NAME?
16  T:
17  A:
18  T:
19  M: OK, GOOD, LOUSY, FINE, NOT BAD
20  MN: NOT SO GOOD, WONDERFUL, TOPS
21  MN: SUPER, IN THE PINK, STUPID
22  T:
23  JY: *END
24  JN: *BEGIN
25  *END
26  END:
```

Example 1
Extending the MATCH command

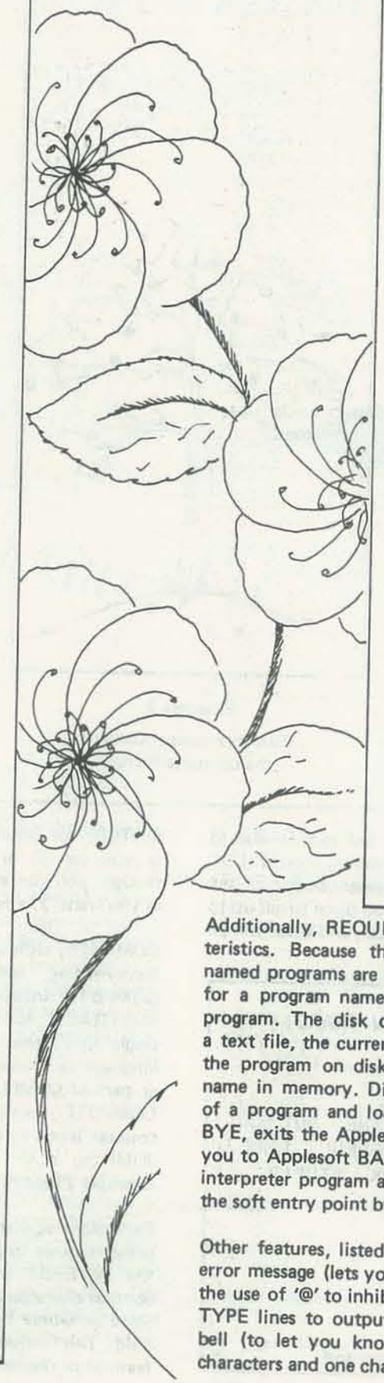
MATCH, for instance, appears to allow only as many items as you can fit on one line. By using the 'N' conditioner, though, you can extend MATCH for as many lines of items as you want. The following example shows one possibility:

COMPUTE, although not too powerful in this version, allows incrementing and decrementing a counter. Also, the COMPUTE instruction can perform the functions ADD, SUBTRACT, MULTIPLY and DIVIDE. You're limited to single digit integers but this is plenty for a non-mathematical language (a future Micro-PILOT will have all BASIC features as part of COMPUTE). A JUMP to a subroutine containing a COMPUTE counter (C+1) instruction will increment the counter (keep a count of correct guesses, for instance). Using JUMP to a C-1 instruction would decrement the count. Example 2 illustrates some possibilities for COMPUTE.

Extensions added to this version of PILOT make the interpreter unique to Apple and more interesting to use. First are the INVERSE and FLASH modes. With the use of three control characters (I, N&F), you can cause any character or word or whole line to be displayed in an inverse or flashing field. This makes it possible to accentuate certain program features or results.

```
REQUEST? RUN

RUN OF APPLE BLOSSOMS

HI...  MY NAME IS APPLE II !
       WHAT IS YOUR NAME ?

?BRIAN

IO YOU WANT TO TRY A VOWEL (V)
OR A CONSONANT (C) ?
TYPE A 'V' OR A 'C'

?V
NAME A VOWEL BRIAN !

?C

THAT IS NOT A VOWEL BRIAN !
NAME A VOWEL BRIAN !

?O


THAT'IS'CORRECT BRIAN !!!
YOU NOW HAVE
C+1 = 1
CORRECT ANSWERS

IO YOU WANT TO TRY IT AGAIN ?
TYPE Y FOR YES AND N FOR NO.

?Y

IO YOU WANT TO TRY A VOWEL (V)
OR A CONSONANT (C) ?
TYPE A 'V' OR A 'C'

?C

NAME A CONSONANT BRIAN !

?I

THAT IS NOT A CONSONANT BRIAN !

!AME A CONSONANT BRIAN !

?X


THAT'IS'CORRECT BRIAN !!!
YOU NOW HAVE
C+1 = 2
CORRECT ANSWERS

IO YOU WANT TO TRY IT AGAIN ?
TYPE Y FOR YES AND N FOR NO.

?Y

IO YOU WANT TO TRY A VOWEL (V)
OR A CONSONANT (C) ?
TYPE A 'V' OR A 'C'

?C

NAME A CONSONANT BRIAN !

?G


THAT'IS'CORRECT BRIAN !!!
YOU NOW HAVE
C+1 = 3
CORRECT ANSWERS

IO YOU WANT TO TRY IT AGAIN ?
TYPE Y FOR YES AND N FOR NO.

?N

THANKS FOR PLAYING BRIAN .
HOPE WE CAN IO IT AGAIN SOON.

BYE...

END OF RUN
```

Figure 1B
Run of Apple Blossoms



```
APPLE BLOSSOMS

 0  R:VOWELS AND CONSONANTS
 1  !
 2   T:HI...  MY NAME IS APPLE II !
 3   T:       WHAT IS YOUR NAME ?
 4   T:
 5   A: #NAME
 6  *BEGIN
 7   T:
 8   T:IO YOU WANT TO TRY A VOWEL (V)
 9   T:OR A CONSONANT (C) ?
10   T:TYPE A 'V' OR A 'C'
11   T:
12   A:
13   M:V,v
14   JN:*CONSONANT
15  *VOWEL
16   T:NAME A VOWEL #NAME !
17   T:
18   A:
19   T:
20   M:A,E,I,O,U
21   TN:THAT IS NOT A VOWEL #NAME !
22   JN:*VOWEL
23   JY:*COUNT UP
24  *CONSONANT
25   T:
26   T:NAME A CONSONANT #NAME !
27   T:
28   A:
29   T:
30   M:B,C,D,F,G,H,J
31   MN:K,L,M,N,P,Q,R
32   MN:S,T,V,W,X,Y,Z,Z
33   TN:THAT IS NOT A CONSONANT #NAME !
34   JN:*CONSONANT
35   T:
36  *COUNT UP
37   T:
38   ! FLASH THIS NEXT LINE
39   T:THAT'IS'CORRECT #NAME !!!
40   T:YOU NOW HAVE
41   C:C+1
42   T:CORRECT ANSWERS
43  !
44  !
45   T:
46   T:IO YOU WANT TO TRY IT AGAIN ?
47   T:TYPE Y FOR YES AND N FOR NO.
48   T:
49   A:
50   T:
51   M:Y,y
52   JY:*BEGIN
53   T:THANKS FOR PLAYING #NAME .
54   T:HOPE WE CAN IO IT AGAIN SOON.
55   T:
56   T:BYE...
57  END:
```

Listing 1
Apple Blossom Program
Vowels & Consonants

Additionally, REQUEST inputs also have some special characteristics. Because the interpreter includes disk commands, named programs are needed. A NEW input to REQUEST asks for a program name. Inputs LIST and RUN use the named program. The disk command DSAVE, saves on the disk, as a text file, the current named program. REPLACE exchanges the program on disk with the current program of the same name in memory. Disk command DLOAD asks for the name of a program and loads the one you name. A REQUEST of BYE, exits the Apple II Micro-PILOT interpreter and returns you to Applesoft BASIC. If you don't do anything else to the interpreter program at this point, you can return to PILOT via the soft entry point by typing (and entering) GOTO2.

Other features, listed back at the beginning, include: Syntax error message (lets you know if you didn't start the line right); the use of '@' to inhibit carriage return (in case you want your TYPE lines to output end-to-end); and a line-limit warning bell (to let you know when your TYPE line is within five characters and one character of the end).

The Apple II Micro-PILOT interpreter also features: LIST suspension with the space bar (stops the program during listing so you can examine it, starts again with any key); control of LIST speed with the position setting of the game paddle control (PDL 0), so you can make it list slow enough to read it ; and a feature that I really like—being able to turn on and off my printer, or any peripheral, with a REQUEST PR# (X). (Where X is the slot number.)

Listing 1 is a trivial program I worked up as an example of something done with this interpreter. The illustrations in Example 2 were done this way, too. I think that PILOT has a lot more potential than shown by these examples. I'm going to look for some of the possibilities.

I'd be interested in exchanging ideas and programs with any and all of you interested PILOTs, too. I've had a lot of fun working with Apple II Micro-PILOT. I am looking forward to helping my youngster learn to program with PILOT—he's almost three now. When the version with extended COMPUTE and USE is ready, I will be attempting some computer-assisted electronics instruction. My plans also include some computer-assisted manufacturing assembly instructions.

If I've stimulated your interest in Apple II Micro-PILOT, you can contact Arley Dealey, High Wizard, at: Magicke Software, 3000 Hood St., Dallas, TX 75219.

The whole set-up at the Carpenter place.

```
**********************
*                    *
* APPLE II MICRO-PILOT *
*                    *
**********************

PILOT FOR APPLE II
IN APPLESOFT BASIC
BY: M.(ARLEY)DEALEY
FEATURE DESCRIPTION
BY: CHUCK CARPENTER

HERE IS A SUMMARY LIST
OF APPLE II PILOT COMMANDS:

INSTRUCTIONS
------------
T: TYPE A MESSAGE
A: ASK A QUESTION
M: MATCH FOR KEYWORDS
J: JUMP TO LABELED LINE
R: REMARK OR COMMENT
C: COMPUTE A RESULT
U:(COMING LATER)
E: END A PROCEDURE

DONE - EXIT THE PROGRAM

CONDITIONERS
------------
Y: CONDITION INSTRUCTION WITH
   POSITIVE MATCH
N: CONDITION INSTRUCTION WITH
   NEGATIVE MATCH

VARIABLES
---------
$=STRING
*=LABEL (SUBROUTINE)
!=ALTERNATE FOR REM OR COMMENT

EXTENSIONS
----------
CONTROL I = INVERSE MODE
CONTROL N = NORMAL MODE
CONTROL F = FLASH MODE

REQUEST?
 NEW - START OF NEW PROGRAM
       ASKS FOR A PROGRAM NAME
 LIST - LISTS NAMED PROGRAM
 RUN - RUNS NAMED PROGRAM
 EDIT - ALLOWS CHANGING ENTRIES
 DSAVE - SAVES NAMED PROGRAM TO
         DISK
 DLOAD - LOADS NAMED PROGRAM
         FROM DISK
 REPLACE - REPLACE PREV. NAMED
           PROGRAM WITH CURRENT
           VERSION
 BYE - EXITS PILOT BACK TO BASIC

OTHER FEATURES
----- --------
SYNTAX ERROR MESSAGES
@ INHIBITS CARRIAGE RETURN
PERIPHERAL CONTROL FROM REQUEST
LINE LIMIT WARNING BELL
SOFT ENTRY AFTER 'BYE'
PDL(0) CONTROLS LIST SPEED
SUSPEND LISTING WITH SPACE BAR

THEMS THE HIGHLIGHTS FOLKS


END OF RUN

REQUEST? REPLACE
```

Apple II Micro-PILOT Features

FORTRAN MAN

Volume III
Episode 7

BY LEE SCHNEIDER & TODD VOROS

As you may recall from our last episode, strange things have been executing in microprocessorland lately . . .

First comes the infamous Glitchmaster, who tries to claim the Land of the Little People as his own! Then comes the Underground Resistance Movement, to pull down the current regime and reload the old order . . . and the battle is on!

Units of the Resistance army, under Linea, move to join General Wirewound in an attack on the Capital City . . . yet as they approach, the drives of their disc transports are neutralized by the powerful Lockout Monster . . . the discs crash, and many of the resistance are taken prisoner!

A stranger amongst the resistance saves them . . . by miraculously reducing the Monster's DIMENSIONS! And this stranger . . . although he himself does not remember who he is (due to an accident in which he lost part of his memory) is none other than . . . FORTRAN MAN!

But time is running out for the Resistance . . . and to regain their strength, they must resort to drastic limits; a jailbreak!

With the assistance of the now-reduced Lockout Monster, they invade FIFO Fortress, and before the Glitchmaster's Guards know what executed past them, the prisoners are back out in free memory space . . . and the *Guards* are locked out!

And yet Our Hero is dejected . . . for still he cannot seem to re-CALL his identity. Then, from out of the lines of recently unterminated resistance a somehow-familiar figure emerges . . . and as F-Man watches in a puzzled manner a highly-activated routine branches in his direction . . .

F-Man! It's me . . . Billy Basic!

I'd almost given up hope of ever locating your line number again . . . How did you escape? What happened to you? Don't you know me?

Eh? I'm sorry, but I'm afraid I don't recognize your pattern!

Are you sure? I know you've told me about him . . . but *this* couldn't be.

And I tell you recursively, he *is*! Where did you find him?

We found him on a PROM, floating on his outputs down the third state of the I/O channel! We reloaded him back into execution . . . but it seems that somehow a portion of his memory is erased!

Linea . . . who is this? Do you know the ID of this segment of code?

Oh yes, comrade . . . Billy and I have known each other for many revisions, ever since he was a little subroutine! We grew up together in Micro-Land!

I should introduce you . . .

WHAT'S HIS LINE?

What? you mean you *know* the filename of this program?

Why, of course! This is the one and only Fortran Man: most famous citizen of 360 City. Fighter of Computer Crime, Corneleus Cobol and the evil Count Algol, Champion of Truth, Justice, and the Algorithmic Way!!!!!

The Bulrush Episode

But, Linea! I already know him!!

Our Hero turns to Linea, who has just arrived to inspect her newly-regained resistance units . . .

The Resistance Commander only blinks once at Billy; then branches quickly towards him.

I am?

It is?

Holy Hollerith . . . of course! That Data Security officer that caught me trying to smuggle him past the Gates . . . he tried to use a U-V Projector on the PROMs! It was only a short exposure, but . . .

short exposure

Then this *is* the real Fortran Man! No wonder he has such powers. I'd *wondered* how he could do all those things.

Imagine that! I'm famous!

The question is, how can we help him?

But Linea! That's just my breakpoint! You've only seen a *few* of F-Man's powers . . . restore his memory space and he could out-compute the Glitchmaster's entire army!

Hmmmmm . . . a possibility. And there just might be a way . . .

As I recall, *you* are the land's greatest authority on Fortran Man . . . correct?

Um . . . as much as I'd like to I don't think we've got the real-time! General Wirewound will still be converging his resistance networks on the Capital City . . . and if *we* are not there to help him, the war could be lost! Already we have lost many valuable cycles . . . I don't know if we can even *get* there in time anymore!

Well, uh . . . I guess so. I must know every detail of every adventure he ever had . . . even those where I wasn't with him! But I don't see what *that* has to do with . . . what are you suggesting?

And even if we do, with my forces weakened as they are . . .

. . . if the mention of its name were a direct . . . L, the creature Linea had mentioned is . . . enly locked in on a startled Billy . . .

. . . ha . . . ???

. . . h . . . er . . . gulp . . . um . . .

. . . llo there?

No time for game-playing now, Billy. Come, we must hurry!

gleep?

streams . . . and as they go there is at last time for inter-transmission of personal data.

You realize how risky this is, Billy. If it doesn't work, your own memory could be erased . . . dissipated forever!

I only hope this plan works, Billy. Microprocessorland depends on you!

I . . . I know, Linea . . . but F-Man has saved my code from destruction more times than I can remember! And besides, to save our homeland . . . I *must* take the chance!

There it is, comrades . . . there stands Castle McIntel, Stronghold of the Clan, Holders of Knowledge and Protectors of the Little People!

Very impressive, Linea . . . but just how do we get across?

Quickly the resistance units accumulate their current supplies, disassemble their camp into relocatable object modules, load their transports, and begin to move. Linea, as always, forms the lead of the Resistance, with Billy at her side and F-Man leading the Monster along on its character string.

They travel onward, for almost a full cycle . . . which in Micro Land is generally longer than most . . . but at last they stand at the falling edge of the Great Voltage Divide.

Careful there, comrades . . . many a good resistance has met their end trying to bridge the Voltage Divide.

Short-circuitly the Divide is spanned and balanced. Across this live bridge the forces of the Resistance propagate towards the great Castle on the opposite side . . .

Why my resistance troops will form a Bridge, of course!

First Decade . . . FRONT AND CENTER!!!!

Come on, everyone . . . no time to lose!

Drawing by A. Miya

Within nanoseconds of the command being issued, units of Linea's precision resistance rush forward and begin to construct and balance the bridge.

Will Linea's plan actually hold up under execution? Will Fortran Man ever recover his memory space and proper ID? Will Billy Basic survive his plug-in operation? Will Micro-Land ever be free of the Glitchmaster? Will this inane comic strip adventure *ever* end?

For the answer to these and other equally obtuse questions, tune in next episode: same cycle, same tolerance band!

FORTRAN MAN

Volume III
Episode 7

BY LEE SCHNEI
& TODD VORC

As you may recall from our last episode, st
things have been executing in microproc
land lately . . .

First comes the infamous Glitchmaster,
tries to claim the Land of the Little P
as his own! Then comes the Undergroun
sistance Movement, to pull down the c
regime and reload the old order . . . an
battle is on!

Linea . . . who is this? Do you know the ID of this segment of code?

Oh yes, comrade . . . Billy and I have known each other for many revisions, ever since he was a little subroutine! We grew up together in Micro-Land!

I should introduce you . . .

WHAT'S HIS LINE?

But, Linea! I already know him!!

What? you mean you *know* the filename of this program?

Why, of course! This is the one and only Fortran Man: most famous citizen of 360 City. Fighter of Computer Crime, Corneleus Cobol and the evil Count Algol, Champion of Truth, Justice, and the Algorithmic Way!!!!!

I am?

Our Hero turns to Linea, who has just arrived to inspect her newly-regained resistance units . . .

couldn't be.

And I tell you recursively, he *is*! Where did you find him?

We found him on a PROM, floating on his outputs down the third state of the I/O channel! We reloaded him back into execution . . . but it seems that somehow a portion of his memory is erased!

The Bulrush Episode

It is?

Holy Hollerith . . . of course! That Data Security officer that caught me trying to smuggle him past the Gates . . . he tried to use a U-V Projector on the PROMs! It was only a short exposure, but . . .

short exposure

Then this *is* the real Fortran Man! No wonder he has such powers. I'd *wondered* how he could do all those things.

Imagine that! I'm famous!

The question is, how can we help him?

Um . . . as much as I'd like to I don't think we've got the real-time! General Wirewound will still be converging his resistance networks on the Capital City . . . and if *we* are not there to help him, the war could be lost! Already we have lost many valuable cycles . . . I don't know if we can even *get* there in time anymore!

And even if we do, with my forces weakened as they are . . .

But Linea! That's just my breakpoint! You've only seen a *few* of F-Man's powers . . . restore his memory space and he could out-compute the Glitchmaster's entire army!

Hmmmmm . . . a possibility. And there just might be a way . . .

As I recall, *you* are the land's greatest authority on Fortran Man . . . correct?

Well, uh . . . I guess so. I must know every detail of every adventure he ever had . . . even those where I wasn't with him! But I don't see what *that* has to do with . . . what are you suggesting?

A Memory-to-Memory transfer, of course! Yours to his! It's only a simple Copy function.

But Linea! That would require the use of an in-circuit emulator, at *least* . . . and surely you don't have *that* kind of equipment out here in the field!

But . . . throughout the history of Microprocessor Land the Clan has always stayed carefully isolated from the rest of the country. Why would they give up their neutral potential and help us *now*?

Because, Billy, we would be returning one of the greatest treasures of their clan – the Lockout Monster!

As if the mention of its name were a direct CALL, the creature Linea had mentioned is suddenly locked in on a startled Billy . . .

Wha . . . ???

Oh . . . er . . . gulp . . . um . . . hello there?

No time for game-playing now, Billy. Come, we must hurry!

gleep?

Of course we don't . . . but Clan McIntel does!

The journey is long . . . yet without PAUSE they execute onward, over the great data fields, through silicon valleys and across serial data streams . . . and as they go there is at last time for inter-transmission of personal data.

You realize how risky this is, Billy. If it doesn't work, your own memory could be erased . . . dissipated forever!

I only hope this plan works, Billy. Microprocessorland depends on you!

I . . . I know, Linea . . . but F-Man has saved my code from destruction more times than I can remember! And besides, to save our homeland . . . I *must* take the chance!

There it is, comrades . . . there stands Castle McIntel, Stronghold of the Clan, Holders of Knowledge and Protectors of the Little People!

Very impressive, Linea . . . but just how do we get across?

Quickly the resistance units accumulate their current supplies, disassemble their camp into relocatable object modules, load their transports, and begin to move. Linea, as always, forms the lead of the Resistance, with Billy at her side and F-Man leading the Monster along on its character string.

They travel onward, for almost a full cycle . . . which in Micro Land is generally longer than most . . . but at last they stand at the falling edge of the Great Voltage Divide.

Short-circuitly the Divide is spanned and balanced. Across this live bridge the forces of the Resistance propagate towards the great Castle on the opposite side . . .

Careful there, comrades . . . many a good resistance has met their end trying to bridge the Voltage Divide.

Why my resistance troops will form a Bridge, of course!

First Decade . . . FRONT AND CENTER!!!!

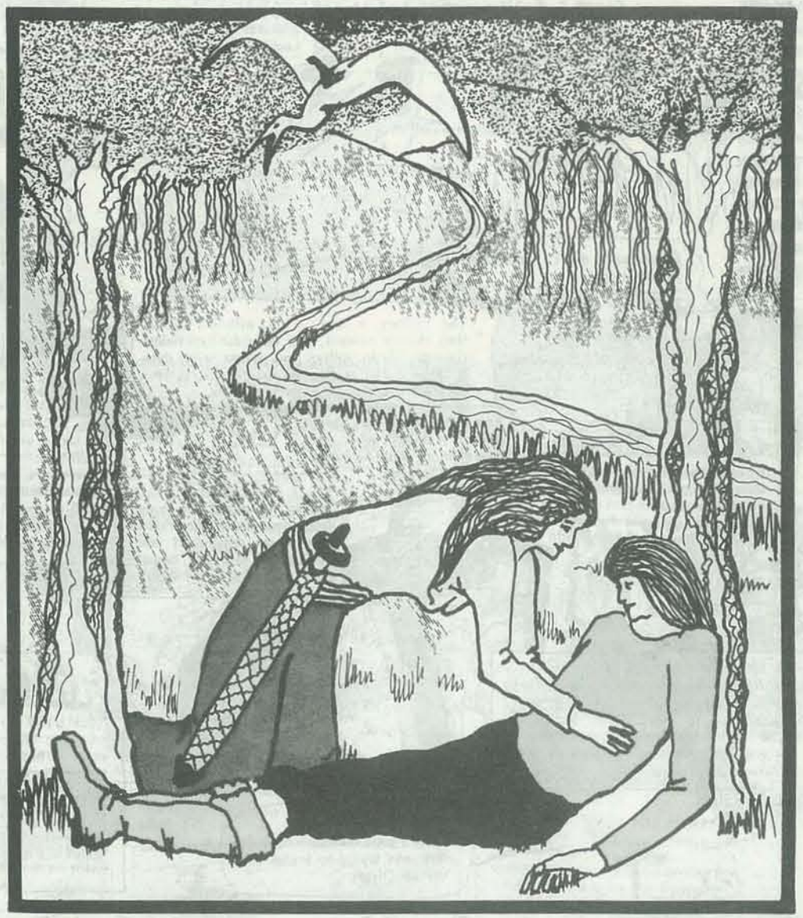
Come on, everyone . . . no time to lose!

Within nanoseconds of the command being issued, units of Linea's precision resistance rush forward and begin to construct and balance the bridge.

Will Linea's plan actually hold up under execution? Will Fortran Man ever recover his memory space and proper ID? Will Billy Basic survive his plug-in operation? Will Micro-Land ever be free of the Glitchmaster? Will this inane comic strip adventure *ever* end?

For the answer to these and other equally obtuse questions, tune in next episode: same cycle, same tolerance band!

What IS All This Stuff?



Beginner's Guide to Fantasy Role-Playing

"What is all this stuff?" Yes, we've heard that more than once, as new RC *readers, paging through the magazine, discover articles talking about fantasy lands, epic games, and role-playing; not to mention, abundant references to dragons. The following article, reprinted from the premier issue of* Different Worlds, *goes a long way toward answering that question. It is one of the clearest, most comprehensive explanations of fantasy role-playing (otherwise known as FRP) we've seen.*

It should also give you an idea of why computer nuts are often attracted to this form of gaming. You want lots of variables and excitement and programs that "learn?" Well, consider the possibilities in FRP. For practical tips on how to apply a computer in the early stages of play, read the Dragon's comments in our continuing series on Runequest, which returns next issue.

Different Worlds *is a new role-playing magazine from The CHAOSium (P.O. Box 6302, Albany, CA 94706). Subscriptions are $9 for one year (six issues).*
—LB

BY CHARLIE KRANK

Few indeed are the days when someone hasn't come into the store and asked, "*What is all this stuff?*" What they were looking at are the miniature figures and the rule systems of fantasy and science-fiction role-playing games. The first of the these was a fantasy game which began about five or six years ago called *Dungeons and Dragons*. Now, there are several very good systems on the market.

Well, I got so tired of trying to explain what all of this means (and only succeeding in making them even more confused), that I decided to write an article for those who have never even heard of role-playing. To do this in sóme sort of logical manner, I have broken the game down into its major components and begin with a definition of what Role-Playing is.

THE DEFINITION

Role-Playing began as an attempt to capture the mystery and adventure of a fantasy world, such as Tolkien's *Lord of the Rings*. A place where magic not only exists but is an accepted practice and even a way of life. Imagine reliving Frodo's trek across Middle Earth or Elric's adventures with Moonglum in the Young Kingdoms. It is also an experiment in the interaction between people. One person (the referee) presents a situation complete with conflict and reward, and the others (the players) try to combine their talents to overcome the challenge and gain the reward. The conflicts usually tend to involve monsters or bad guys, but can also include riddles, traps or mazes. Though the rewards often are a form of treasure (gems, gold or magical items), they can be more abstract, such as solving that riddle, figuring out the trap or making it through the maze.

If the players survive the adventure, they have some method for the advancement of their characters – a form of growth process. As the characters advance, they experience an increase in their fighting skills usually accompanied by an increase in the amount of money found on adventures. Players will use this money to replace, improve or augment their equipment. In a complex world, they may also be required to buy food and lodging.

Role-Playing is not a competitive type of game like chess or the standard wargames, but is rather a cooperative effort on the part of the players to defeat the referee's monsters. Also, there is no true "win" in the game. Each adventure in the game builds on the one before, almost as if each adventure were a chapter in a book. If the player's character survives, he will participate in the next adventure with a more powerful character. If not, he must start over with a new character.

At this point, it may be helpful to look at just what the referee and players do in the game and how they go about resolving the conflicts.

THE REFEREE

In any Role-Playing system, one person has the job of the referee, and serves several purposes. First, it is his responsibility to create the world in which all of the action in the game will take place. The complexity of this world can vary greatly. It can be as simple as a couple of rooms below a castle or as complex as a whole planet, complete with history, continents and a multitude of ongoing adventures.

It all depends on the referee's ambition and the amount of time available to work on it. One very popular time-saving device is to borrow a framework and history from some existing mythos. Of course one of the most popular is J.R.R. Tolkien's *Lord of the Rings* trilogy. Other popular works include Howard's *Conan*, Moorcock's *Elric*, Le Guin's *Earthsea* trilogy and Leiber's *Lankhmar*. CHAOSium has its own fantastic world of *Glorantha*.

Once the referee has determined this structure, he then populates the different areas, be they dungeon rooms that he's drawn out on paper, or the wilderness areas located on his maps, with monsters. The term "monster" is used here and in many rules systems to indicate the animals, humans, and human-type creatures in the world. This would include, for example, the friendly Elven Magic User who, for a price, will show the adventurers out of the area they have gotten lost in.

Certain common-sense types of guidelines should influence the referee's placement of monsters. Large dragons, for instance, would need an immense room if they were not to feel cramped (remember that they do have to stretch their wings at times). Also, certain monsters just naturally do not get along well together. Just a little time spent considering these factors will add incredibly to the believability of play.

The next task of the referee is to place the treasure in his adventure. Usually, when a party runs into a room, almost gets killed, but does succeed in dispatching the monster, the group expects to find a good deal of treasure. I, however,

find it more stimulating when the amount of booty is just enough to pay expenses until the next adventure and possibly replace or improve my weapons and armor.

This type of campaign style serves several purposes: first, it provides continued motivation to adventure; second, it vastly increases the value of plate mail and finely made weapons and, finally, it helps prevent the players from acquiring an arsenal of super-powered goodies and aids with which they breeze through hordes of baddies without the slightest danger to themselves. The final decision will be up to you, but keep in mind that part of the attraction of a game such as this is the struggle to survive and the uncertainty involved in accomplishing that survival.

A third purpose of the referee is to run all of those monsters which were so thoughtfully scattered about. This will be the closest you get to actually playing in your world. The more life that you can give to those beasties, the more enjoyable will be your game to the players. There will be many times that you will develop a kind of attachment to one of your human or inhuman monsters, but one cold, hard fact that every referee must face is that all your creatures will eventually die (that doesn't mean that they won't take an adventurer or two down with them, though). I'm not saying that you should go out and purposely kill off the characters, for if the players feel that that is your whole motivation, then they may stop playing in your world (and all of your work is down the tubes). Instead, a good referee will play the monsters so as to give the greatest challenge to the players. This will keep the game lively and interesting, and a good deal of fun for all parties involved.

A final purpose of the referee is to answer the multitude of questions that the players will ask. Some will be relatively easy, such as, "What are the chances of my character with a dexterity of 17 making a 10-foot jump onto the back of that orc?" Then will be the times when they ask, "Does a Protection from an Evil Spell apply to an animal who is instinctively protecting his territory from intruders? He may not necessarily be considered evil unless he was sent here purposely to harm us but . . . "

Simpler questions on ability can usually be resolved by a die roll. The more complex questions will require some judgment on your part. If you really cannot decide, the players always have suggestions, not all of which can be mentioned in public. You may want to listen to them, but the final decision will have to be yours. Remember also that what is good for the players is good for the monsters, and vice versa. As you become more experienced, you will find that your game will attain an individuality and style all its own and that the players will be eager to find out if they can master its murky depths.

THE PLAYERS

Before the game starts, each beginning player will generate one or more characters who will participate in the adventure. Players who already have characters will just use the ones they have. Each rule system has its own prescribed method for this determination of characters, but they all have certain points in common. The first step will be to generate scores for certain characteristics, such as Strength, Intelligence, Power, Constitution, Dexterity and Charisma.

These scores will provide both an indication of how your character will act in certain situations, and help determine the profession he should follow. If your character has very good strength, for example, he will be a better fighter and will perform better in strength-related activities such as opening locked doors than someone whose abilities lie more in intelligence. The very intelligent character, on the other hand, will have more of a mastery of languages and spells. Both could accomplish the same end, but use different methods.

The next determination will be the assessment of the amount of damage that your character can take before he is killed. Many systems have this linked very closely with the constitution of the character. During the game, the player will have to keep a running total of his character's hit points. If these are exceeded, then the character is dead. It becomes, therefore, very important to protect your character as well as possible. One of the primary ways of doing this is to buy armor. First, though, you must have some amount of money.

The money with which you start the game can be determined in several ways. Some systems use tables showing different social classes and the probabilities for each, with the classes each having amounts of money available. Others use a simple die roll. However it is done, an amount of money will be allotted. With this, you must purchase weapons, armor (as mentioned above) and supplies. The choice of weapon will be basically up to you.

Some systems have restrictions based on character classes (occupations), social ranks or abilities. The different weapons will cost varying amounts of money and will do different amounts of damage. It is often a wise practice to carry more than one if possible, in case you happen to lose or break one. Armor may also have certain restrictions similar to weapons, but will be more influenced by your intended activities and pocket book. While plate mail will give you about the best protection around, it generally costs a great deal and is not ideal for swimmers. Conversely, the lighter forms of armor, leather and chain, allow a good deal more movement and silence, but don't afford as much protection.

Another use for money will be the purchase of accessory equipment. Unless you can see in the dark, as some races can, then torches are a good idea. Of course you will need to buy provisions and some sort of carrying device. Other things such as rope, stakes and mallet, flasks of oil and the like can also be very useful, but their purchase will have to depend on your monetary situation.

Also remember that money is good for the buying of services, repair of armor and weapons, bribing of officials and so forth.

If things are really tight, you could borrow from the town money-lender (at outrageous prices, of course), but remember that you should find at least a little treasure during the course of an adventure, so don't despair. Also, if your character survives the adventure, his fighting skills will develop. When he becomes good enough, he can begin to hire himself out. The number of ways in which money can be made (or acquired) are only limited by your imagination.

Throughout all of this, you will find that your character will become more and more a part of you. He will begin to develop a personality of his own, and increasingly, you will find that you play the character as an individual. This is the essence of Role-Playing, and one of the prime reasons why it has caught on in the past few years. You are able to live out your fantasies through your character, and his death could result in a real sense of loss.

THE RULE SYSTEM

The most important part of any Role-Playing game is the rule system. At this moment, there are quite a few systems on the market. Here I will present only a few. In later articles, we will try to provide a more complete listing.

If you are more interested in fantasy Role-Playing, you might go to your store and look over these systems. From Tactical Studies Rules (TSR) comes the aforementioned *Dungeons and Dragons* and *Empire of the Petal Throne*. A group in Arizona called Flying Buffalo produces one of the more light-hearted systems, called *Tunnels and Trolls*, and for those of you interested in a good deal of historical accuracy concerning the Middle Ages, look at *Chivalry and Sorcery* from Fantasy Games Unlimited (FGU). FGU also makes a game based on the very popular novel *Watership Down*, a book about a rabbit society, named *Bunnies and Burrows*. Game Designers' Workshop has a game which also closely simulates the mood in the age of honor and chivalry called *En Garde*. Legacy Press manufactures *Legacy*, and CHAOSium recently brought forth *Runequest!*

For those of you more into the science-fiction type of adventure, consider *Metamorphosis Alpha*, a trip through a lost spaceship, by TSR. If you would rather run your own ship, give *Starships and Spacemen* by FGU a try, or even their *Flash Gordon* game. GDW makes another spaceship running game called *Traveller*, and Tyr Gamemakers makes an all-encompassing set of spaceship rules called *Spacequest*. If you always wanted to be a superman, try *Superhero 44* from Lou Zocchi. Finally, if your interests lie more in the Wild West framework, look at *Boot Hill* by TSR or *Wild West* from Lou Zocchi.

When looking for a system to use, talk with the people at the store. They may be able to help you decide which system would best suit you. Then, go home and read the rules through several times before playing, or, better yet, try and find somebody who plays the rules. High schools and colleges are often good places to look.

Once you decide on a system, there are several ways to play. Some people like to play orally. The referee describes the rooms, and the players tell him what they are doing. I prefer to use little 25mm lead figures available in many game stores. They help both the players and the referee visualize the action and greatly aid in determining the distances between the players and the monsters. When using these miniatures, the referee will have to indicate the hallways and rooms of his scenarios. Any method is acceptable, be it toothpicks on a tabletop, chalk on a blackboard or grease pencil on plexiglass, as long as all players understand the scale and the system.

THE RESULTS

Finally, there are several effects of becoming involved in Role-Playing. When I first began, I was buying any and everything that I could get my grubby little hands on (my father never could understand how you could spend so much money on just one game). As a result, I am now the proud owner of hundreds of miniature figures, and enough rule systems to start a small store. You will find yourself staying up to all hours of the night devising ways to subtly eliminate the players. All your free time will disappear, and you become very, very poor. Your games will last through one night and on into the next when you drop from exhaustion, only to awake and begin again. You can always tell a referee by the distinctly "undead" look about him. It's great!

THE FINAL NOTE

If you have any questions or comments concerning a rule system or interpretation of rules, please send them in to *Different Worlds*, care of *Beginner's Brew*, P.O. Box 6302, Albany, CA 94706.

Many of the articles on FRP use various abbreviations that are hobby standards. The following is a list of the more common ones.

D4	a four-sided die
D6	a six-sided die
D8	an eight-sided die
D10	a ten-sided die (a twenty-die numbered one to ten twice)
D12	a twelve-sided die
D20	a twenty-sided die
D100	a roll of two D10s to produce random numbers from one to one hundred
D3	a roll of a D6 with results of 1-2=1, 3-4=2, and 5-6=3
3D6	a sum resulting from a roll of three D6s
10xD6	ten times a roll of D6
100x10D10	one hundred times a roll of 10D10
FRP	fantasy role-playing game
RPG	role-playing game
GM	gamemaster
DM	dungeon master
APA	Amateur Press Association

SOFTWARE REVIEWS FROM

THE BEST OF THE PET GAZETTE

The PET Gazette, *published six times a year, contains a wealth of information for PET users. It is FREE!! The address of the publication is 1929 Northport Dr., Room 6, Madison, WI 53704.*

The Best of The PET Gazette *for 1978 recently arrived, and it's a bargain ($10) for all the information it contains. You'll find ads for PET products, software, reviews, programs and much more in the packed 96 pages. Len Lindsay, the editor, notes in the introduction that the BOTPG (Huh?!) contains lots of reviews. He also reminds the reader that if the product is reviewed, then it exists . . . seems there have been problems with people advertising products that don't exist. Len cautions in his introduction: "Never order anything unless you know it exists! If it is reviewed in* The Pet Gazette *you can be sure it exists!" Sounds like good advice. With Len's permission, we have extracted several software reviews from the* BOTPG. *We only chose the reviews that gave products a high rating. Sort of a best of the* BOTPG, *so to speak . . .* — *RZ*

VIDEO CHECKERS (No price listed)
Compu-Quote
6914 Berquist Ave.
Canoga Park, CA 91307

While checkers is certainly not one of the world's most interesting or complex games, I do feel this program is excellent. It is truly a *video* checkers game; the board is POKEd, not PRINTed. In addition, pieces blink when they are moved, making the program easier to play. The program catches illegal moves and will force you to jump a piece if you can. It is the best checkers game I have ever seen. While it does not play a very good game of checkers, it plays better than most. If checkers is your game, then by all means, get this program. (NOTE: I'm impressed with the company — they recorded the program on both sides of the tape and provided fairly decent instructions. Other software vendors take note!)

Reviewed by Jon Staebell

STAR FIGHTER/ASTEROID ($10)
ZZYP Data Processing
2313 Morningside
Bryan, TX 77801

STAR FIGHTER is a STAR WARS simulation. The screen lights up with the sights of your laser gun. You steer your ship to get the enemy fighter in your sights and fire! If you hit the ship, it explodes in an amusing animated scene. In *ASTEROID*, you try to maneuver your space ship up through a group of horizontal moving asteroids. It is not easy to win, but not too discouraging. Again, if you're hit, there is an animated explosion. Both of these games are exciting, easy to use, graphic, and addictive. Best of all, complete documentation is included. AND, they even include a listing of the machine language program used to update the graphics quickly. WOW! A must buy . . .

CASINO ROULETTE ($9.95)
CASINO CRAPS ($9.95)
CMS
5115 Menefee
Dallas, TX 75227

CASINO ROULETTE
This is a great version of roulette. It is designed not only to play the game but to teach it, just the way it is played in a casino. The documentation is complete in every detail.

The displays show the layout of the betting table, the 12 types of bets and their odds.

When you "spin" the wheel, an X moves from number to number on the betting table, stopping by the winning number.

The results of all the bets are shown and your purse is displayed. You then can play again or stop.

CASINO CRAPS
This is the best version of craps we've seen. It has all the betting options of a real casino game. The documentation is extensive and not only tells what the different options are, but what the house odds are against you.

A tremendous game if you want to learn to play craps the way they do in casinos.

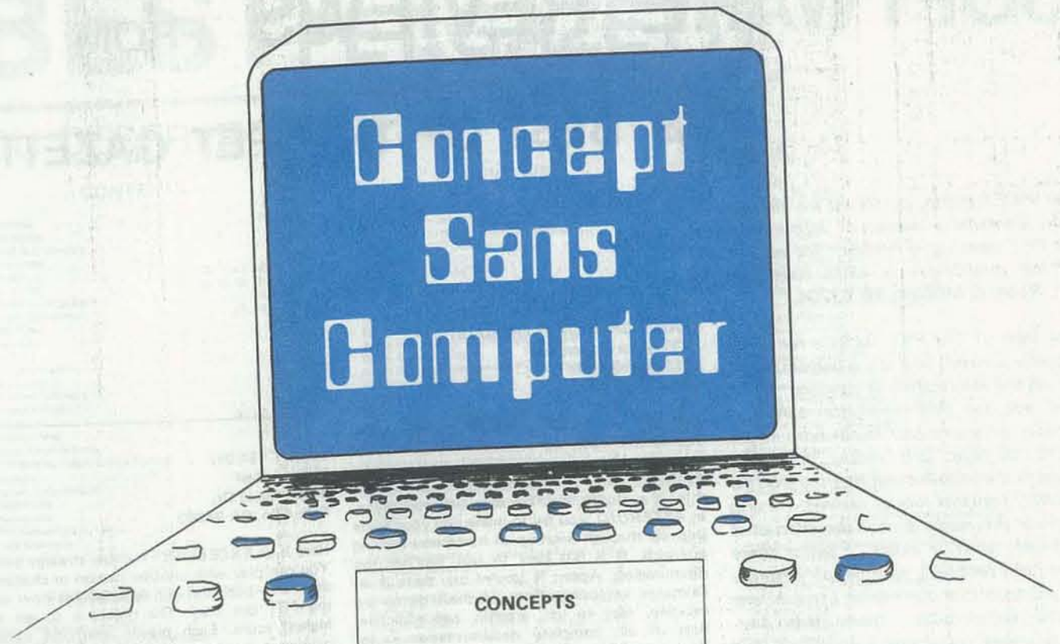
Reviewed by Bill Bendoritis

MAXIT ($4.95)
Harry J. Saal
810 Garland Dr.
Palo Alto, CA 94303

Here is an EXCELLENT number strategy game. You can play with another person or challenge the PET—but you will be surprised how well the PET can play. The object is to get the highest score. Each player alternates moves, picking a number and adding up the points. One player can only choose a square not yet taken in the same row as the marker for that turn; the other player can only choose a square in the same column as the marker. The results are interesting, especially at the end game. A good exercise for thinkers.

SWORDQUEST (No price listed)
Fantasy Software Games
P.O. Box 1683
Madison, WI 53701

SWORDQUEST is the only fantasy simulation I have seen with graphics, animation and (soon) sound effects. The program itself is complex and has several machine language routines. It is well human engineered and easy to play — but also challenging and exciting. You wander about a maze of tunnels, trying to find the room with the treasure. You can kill any monsters with your arrows (except for Giant Spiders — they need a Magic Arrow to assure a kill). The maze of tunnels is always shown on your screen; there is no scrolling. All monsters are invisible until you come within 4 steps of them. They then appear and charge at you, moving 3 spaces each turn. You can do two things for your turn, including move, shoot an arrow, and change weapons. You must decide to carry either your bow or your sword. You must use your sword to fight any monster that attacks you. The intricacies of this game are explained in the accompanying manual. The manual also relates the background story to the game. An amazing aspect of this well designed game is that it is protected; you cannot make a duplicate copy. (Unauthorized copies are illegal anyway.)



Concept Sans Computer

BY ERYK VERSHEN

In the January-February 1979 issue of RC, we presented a Concept Game program for the TRS-80. Almost before the ink dried, Eryk appeared with this clever way to completely solve the game . . . without a computer! Heresy! Stone the infidel! What will happen to this world if people go around thinking? —*RZ*

The Concept Game is rather easy to find solutions for if one has a computer or some patience. Not possessing the former, I relied on the latter and generated the accompanying Venn diagrams. The lines within the diagram gather sets of common Concept Game properties together. These common sets are the solutions to the game.

The notation in the diagram for the various concepts are: *p* for parity, *b* for balance, *m* for majority, *c* for closure, *s* for skip and – for properties which are absent. Certain results are immediately obtainable—no set of properties has more than ten solutions; only two sets (*-bm--* and *pb--s*) have no solutions.

PARITY: Even number of 1's.
Example—1011010

BALANCE: Same number of 1's on each side of the middle digit.
Example—1010011

MAJORITY: More 1's than 0's.
Example—1110011

CLOSURE: First and last digit are the same.
Example—0100110

SKIP: "101" pattern somewhere in the line.
Example—0101011

A general strategy for playing the game combines two techniques—memorization and *proofs*. You use memorization for easy ones like *-bm--* and *proofs* for others. A *proof*, in general, goes as follows: the presence or absence of parity breaks the number of ones in the possible solutions into two sets ($\{0,2,4,6,\}$ and $\{1,3,5,7\}$); the presence or absence of majority causes further subdivision ($\{0,2\}$, $\{4,6\}$, $\{1,3\}$, and $\{5,7\}$); then using the state of balance and closure, it is possible to create general patterns, modified by whether skip is present, to obtain final solutions.

For example: Find the solutions to *p-m-s*.

1. Parity is on so there are either 0, 2, 4 or 6 ones in each answer.
2. Majority is on so there are 4 or 6 ones in each answer.
3. Skip is on and closure is off so 6 ones are not possible.
4. Balance is off and skip is on so the only patterns allowed are 4 ones which preserve this last set of conditions. Balance off means that there must be 3 ones on one side and 1 one on the other with a zero in the center, or 2 ones on one side and 1 one on the other with 1 one in the center. In short, the patterns must be of the form 103 and 112 and the mirror images. Expanding the short notations into the Concept Game solution form gives:

0010111	1110100	103 & mirror image
0011011	1101100	112 & mirror image
0011101	1011100	112 & mirror image
0101011	1101010	112 & mirror image
0101101	1011010	112 & mirror image

p----	pb---	-b---		-----
0000011	0010001	0011001		0000001
0001001	0100001	1001100		0000111
1001000	1000010			0010011
1100000	1000100			0100011
				0110001

p-m--		pbm--	-bm--	--m--	
0001111	1001110	0110011		0011111	1000000
0100111	1110010	1100110		1111100	1000110
0111001	1111000				1100010
0111111	1111110				1100100
					1110000

p-mc-	pbmc-	-bmc-	--mc-	---c-
0011110	1100011	0111110	1001111	0000010
0111100		1111111	1100111	0000100
1000111			1110011	0001110
1110001			1111001	0010000
				0100000

p--c-	pb-c-	-b-c-		
0000110	0000000	0001000		0100110
0001100	0010010	0011100		0110010
0011000	0100010	1001001		0111000
0110000	0100100			1000011
	1000001			1100001

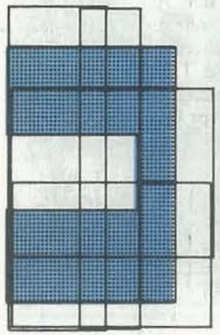
p--cs	pb-cs	-b-cs	--mcs	---cs
0001010	0010100	0011010	1010111	0010110
0101000		0101010	1110101	0110100
		0101100		1000101
				1010001

p-mcs		pbmcs	-bmcs		
0101110	1011111	0110110	1011011		
0111010	1101001	1010011	1011101		
1001011	1101111	1010101	1101011		
1001101	1111011	1100101	1101101		
1011001	1111101	1110111			

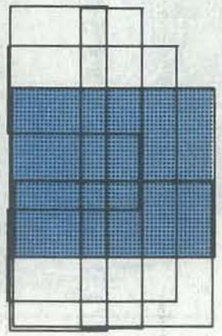
p-m-s		pbm-s	-bm-s	--m-s	----s
0010111	1011010	0110101	0111011	0101111	0001011
0011011	1011100	1010110	0111101	0110111	0001101
0011101	1101010		1011110	1110110	0010101
0101011	1101100		1101110	1111010	0100101
0101101	1110100				
					1010010
					1010100
					1011000

p---s	pb--s	-b--s		
0000101		0101001		1101000
1010000		1001010		

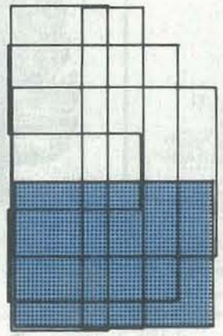
All majority solutions



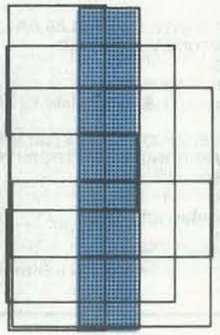
All closure solutions



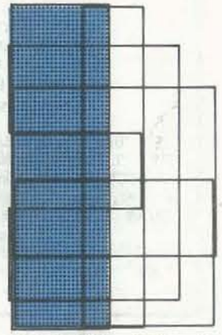
All skip solutions



All balance solutions



All parity solutions



Inspector Clew-So

BY RONALD J. CARLSON

For you budding detectives, here is a game to test your powers of deduction. However, the game does present some difficulties. The suspects are Bill, Mary, Suzy, John and Paul. With names like those, they all sound guilty. – *RZ*

Inspector Clew-So is a computerized detective simulation loosely patterned after the detective board games. However, there are several unique and challenging twists in this game.

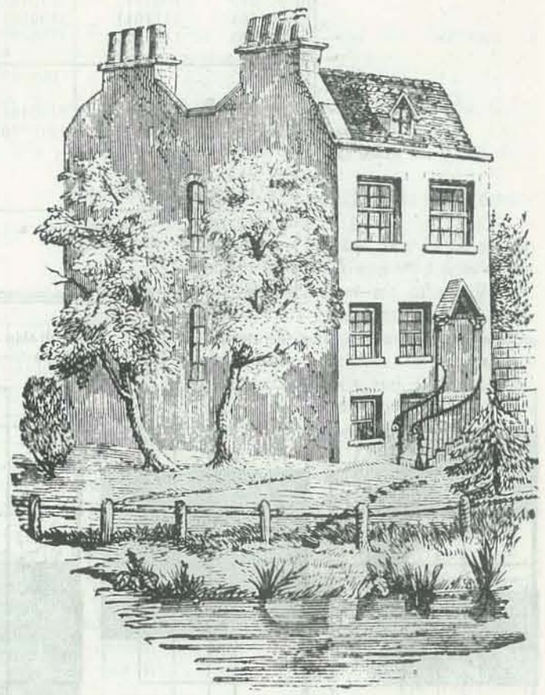
The program was written in BASIC and run with Digital Group MAXI BASIC. Only standard BASIC statements were used to insure portability to other versions of BASIC.

Even if the game is not played with hard copy, the user will need pencil and paper to keep track of the times, places and alibis of the suspects in the house.

A murder has occurred in the guest house. One of the guests (random) has killed the host during the time 1-9 p.m. (random). The great homicide detective, Inspector Clew-So, is allowed to ask the suspects, Bill, Mary, John, Suzy, or Paul, for their location in the house at a particular time. The suspects will answer and also say who was with them and who they saw in adjacent rooms.

As a further check, or as a different approach, the inspector may ask the suspects, at what time(s) they were in an individual room. The suspects move from room to room each hour. The guilty person will lie (randomly) about his/her whereabouts and the condition of the victim.

The inspector must collect and analyze enough answers to determine who is lying and thus, the guilty person. Then the inspector has to narrow down the location and time of the murder. When the inspector has part of the crime solved, (suspect, room, or time), this may be confirmed or rejected with a direct confrontation. If the inspector is completely flabbergasted and resigns, then the facts that eluded the inspector during the questioning are displayed.



MAJOR VARIABLES

C	Keeps track of the number of questions
C1	Keeps track of the number of confrontations
P(5,9)	Represents the position in the house for all five suspects for the hours 1-9 p.m.

THE NEXT THREE VARIABLES ARE RANDOMLY ASSIGNED

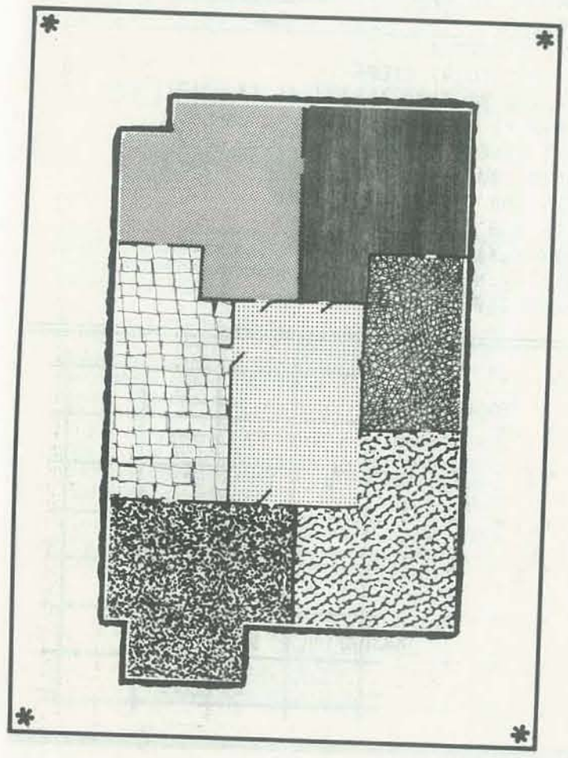
M	Killer (1-5)
T	Time (1-9) of the murder
R	Room location (1-6), determined by P(M,T)

S$ = "BILLMARYPAULSUZYJOHN" . . . 5 four letter names
R$ = "LOUNGEATRIUMLIVINGDININGTROPHYGARAGE"
. . . 6 six letter rooms

FNA$	Suspect number	→	Name
	or		
	Room number	→	Name
FNB	Name	→	Suspect number or room number

Listing

```
10DIM S$(20), R$(36), A$(3), P(5,9), Z$(36), Y$(36) , H$(36)
20REM**************************************
30REM*                                    *
40REM*     INSPECTOR CLEW-SO              *
50REM*     BY: RON CARLSON                *
60REM*     FEB. 1979                      *
70REM*                                    *
80REM**************************************
90S$="BILLMARYPAULSUZYJOHN"
100R$="LOUNGEATRIUMLIVINGDININGTROPHYGARAGE"
110DEF FNA$(M$,P,O)=M$((P-1)*O+1,P*O)
120DEF FNB(Z$,Y$,L,H)
130FORA=1TOH
140IFZ$<>FNA$(Y$,A,L) THEN170
150P=A
160EXIT190
170NEXTA
180P=0
190RETURNP
200FNEND
210PRINT"BILL, MARY, JOHN, SUZY AND PAUL ARE HOUSE GUESTS.THEIR HOST "
220PRINT"WAS MURDERED BY ONE OF THEM BETWEEN 1 PM. AND 9 PM."
230PRINT"YOUR JOB AS INSPECTOR CLEW-SO, IS TO FIND THE KILLER,TIME & ROOM."
240PRINT"YOU WILL BE GIVEN A HOUSE DIAGRAM AND A SET OF QUESTIONS"
250PRINT"FOR THE SUSPECTS,BUT THE GUILTY PERSON MAY TRY TO MISLEAD YOU."
260PRINT"BY LYING SOME OF THE TIME."
270PRINT"IF ONE OF THE SUSPECTS CLAIM THAT THE HOST WAS ALREADY DEAD,"
280PRINT"OR THAT THE HOST WAS STILL ALIVE, THEN YOU HAVE FOUND THE"
290PRINT"ROOM WHERE THE MURDER TOOK PLACE."
300 PRINT
310H=0
320C=0
330C1=0
340PRINT"   |-------------------------|"
350PRINT"   |           |  O GARAGE   |"
360PRINT"   |  O TROPHY |             |"
370PRINT"   | DINING |--|--  --       |"
380PRINT"   |        |     |  |       |"
390PRINT"   |--O-----|     |--|  O (WINDOWS)"
400PRINT"   |                |       |"
410PRINT"   |          |-----  -------|"
420PRINT"   | LIVING   |--|           |"
430PRINT"   |--        |  |           |"
440PRINT"   |     O ATRIUM |  O LOUNGE |"
450PRINT"   |--------------|-----------|"
460FORK=1TO5
470P(K,1)=INT(6*RND(K))+1
480NEXTK
490FORL=2TO9
500FORK=1TO5
510A=INT(6*RND(K))+1
520IFA=P(K,L-1)THEN510
530P(K,L)=A
540NEXTK
550NEXTL
560REM ESTABLISHED SUSPECT'S MOVEMENTS
570 REM RANDOM ASSIGNMENT OF KILLER,TIME AND ROOM
580M=INT(5*RND(1))+1
590T=INT(9*RND(2))+1
600R=P(M,T)
610INPUT"INSPECTOR CLEW-SO WHO IS YOUR SUSPECT ? ",S1$
620S=FNB(S1$,S$,4,5)
630IFS=0THEN610
640PRINT
650PRINT"DO YOU WISH TO QUESTION "#S1$;" ABOUT :"
660PRINT"   1   - THE SUSPECTS WHEREABOUTS AT A PARTICULAR TIME"
670PRINT"   2   - WHAT TIME THE SUSPECT WAS IN A CERTAIN ROOM"
680INPUT"   3   - THE CRIME IS SOLVED ---",A
690IF A<1 OR A>3 THEN650
700C=C+1
710ON A GOTO 720,990,1200
720PRINT S1$,
730 REM TIME SECTION
740INPUT "WHERE WERE YOU AT ",T1
750IF T1<1 OR T1>9 THEN740
760R1=P(S,T1)
770IFS<>M THEN860
780 REM LIAR SECTION
790IFRND(2)<.5THEN910
800R1=INT(6*RND(3))+1
810IF RND(4)<.5 THEN 840
820PRINT"THE HOST WAS ALREADY DEAD."
830GOTO910
840PRINT"OUR HOST WAS STILL ALIVE."
850GOTO910
860IFR1=RTHEN880
870GOTO910
880IFRND(5)<.5 THEN910
890IF T1 < T THEN PRINT" THE HOST WAS STILL ALIVE."
900IF T1 > T THEN PRINT" THE HOST WAS ALREADY DEAD."
910PRINT"I WAS IN THE ";FNA$(R$,R1,6)" ROOM."
920FORK=1TO5
930IFK=S THEN960
940IF P(K,T1)=R1 THEN PRINT"I WAS WITH ";FNA$(S$,K,4)
950IF ABS(R1-P(K,T1))=1 THEN PRINT"I SAW "FNA$(S$,K,4)
960NEXTK
970GOTO610
980REM ROOM QUESTIONING
990PRINT S1$,
1000INPUT " WHAT TIME WERE YOU IN [ROOM] ",R1$
1010T1=FNB(R1$,R$,6,6)
1020IFT1=0THEN1000
1030IFS<>M THEN1110
1040IFRND(5)<.5THEN1110
1050T1=INT(RND(5)*6)+1
1060GOTO1090
1070IF T1=F THEN PRINT"I WAS NOT IN THAT ROOM."
1080GOTO610
1090PRINT"I WAS IN THAT ROOM AT ";T1
1100GOTO610
1110K=0
1120FORB=1TO9
1130IF P(S,B)<>T1 THEN 1160
1140PRINT"I WAS IN THAT ROOM AT ";B
1150K=1
1160NEXTB
1170IF K=0 THEN PRINT"I WAS NOT IN THAT ROOM."
1180GOTO610
1190REM CONFRONTATION SECTION
1200C1=C1+1
1210PRINT"INSPECTOR DO YOU THINK YOU KNOW :"
1220PRINT"  1   KILLER"
1230PRINT"  2   ROOM"
1240PRINT"  3   TIME"
1250INPUT"  4   TOTALLY BAFFLED ---",A
1260 ON A GOTO 1270,1420,1350, 1550
1270INPUT" THE KILLER IS ? ",S1$
1280X=FNB(S1$,S$,4,5)
1290IFX=0THEN1210
1300IF S1$<>FNA$(S$,M,4) THEN 1530
1310PRINT" YOU HAVE THE KILLER, INSPECTOR CLEW-SO."
1320H=H+1
1330IF H=3 THEN 1500
1340GOTO610
1350INPUT "TIME OF THE MURDER ",T1
1360IF T1<1 OR T1>9 THEN1350
1370IF T1<>T THEN 1530
1380PRINT"INSPECTOR YOU HAVE THE RIGHT TIME."
1390H=H+1
1400IF H=3 THEN 1500
1410GOTO610
1420INPUT "ROOM OF THE MURDER ",R1$
1430X=FNB(R1$,R$,6,6)
1440IFX=0THEN 1420
1450IF R1$<>FNA$(R$,R,6)THEN 1530
1460PRINT"INSPECTOR , YOU NOW HAVE THE ROOM."
1470H=H+1
1480IF H<>3 THEN 610
1490 REM CONFIRMATIONS
1500PRINT"YOU ARE BRILLIANT INSPECTOR CLEW-SO."
1510PRINT"IT TOOK YOU "C1" QUESTIONS AND "C1"CONFRONTATIONS."
1520GOTO1590
1530PRINT"INSPECTOR CLEW-SO YOU ARE A BUMBLING IDIOT.TRY AGAIN"
1540GOTO610
1550PRINT"TOO BAD INSPECTOR CLEW-SO"
1560PRINT" THE FACTS ARE:"
1570PRINTFNA$(S$,M,4)" KILLED THE HOST AT "T1" O'CLOCK IN THE "
1580 PRINT FNA$(R$,R,6)" (ROOM)"
1590INPUT"DO YOU WANT A NEW CASE INSPECTOR ? ",A$
1600IF A$="YES" THEN 210
1610END
READY
```



test pattern

BY MILAN D. CHEPKO

For the readers new to RC, *Milan is an MD who lives in Thief River Falls, Minnesota. He is an ardent TRS-80 user and article writer. Expect to see more from him in the months to come . . . –RZ*

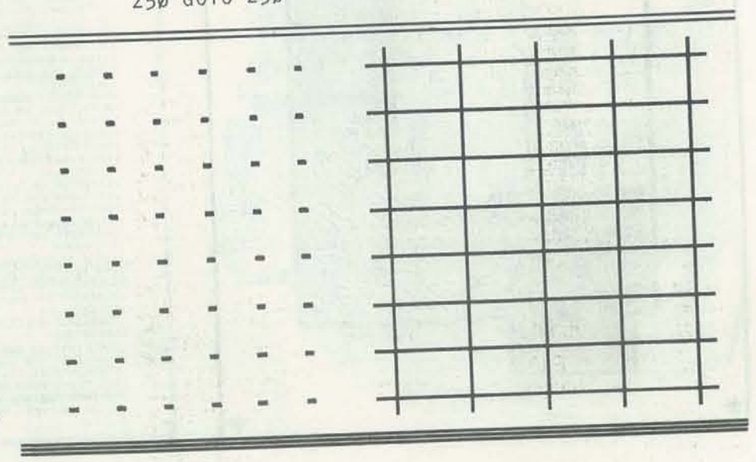
In order to adjust the picture on a TV, a stable, repetitive pattern is essential. For many of us, the need to make such adjustments only comes up occasionally, so the investment in a professional pattern generator cannot be justified. However, a microcomputer can substitute if a program is written to provide such a pattern. A repeating pattern of characters can be used, but the graphics capabilities of the TRS-80 make possible a better simulation of the traditional dot and line patterns.

The program itself is quite simple. Lines 10-40 draw a series of vertical lines, and lines 50-80 draw the horizontal lines; the program then enters a loop until you hit BREAK on the keyboard. The dot pattern is generated by lines 200-230, and again a loop is used to keep the pattern on the screen. The number and position of the lines and dots can easily be changed to suit your needs.

While it does not seem to be generally known, the TRS-80 will work fine with an RF generator and normal TV as its monitor. The Users Manual shows on page 228 that pin #4 carries the video signal, with pin #5 as the ground. If you buy an extra 5-pin DIN plug, just solder the center wire of a length of RG-58U coax to pin #4 and the shield to pin #5, and connect the other end to the input of the RF generator. I found that the resulting picture was quite poor until I soldered a 5 ohm resistor across pins 4 and 5. Depending on the type of generator you use, the resistor may not be needed.

Listing

```
LIST
 1 CLS:PRINT"  TV ALIGNMENT PROGRAM"
 2 REM  BY MILAN D. CHEPKO, M.D.
 3 REM  THIEF RIVER FALLS, MINN.  56701
 5 PRINT:PRINT"THIS PROGRAM WILL DRAW
   STABLE PATTERNS"
 6 PRINT"SO YOU CAN ADJUST YOUR TV."
 7 PRINT:PRINT"TYPE '1' FOR
   INTERSECTING LINES"
 8 PRINT"TYPE '2' FOR DOT PATTERN"
 9 INPUT X:ON X GOTO 10,200:GOTO9
10 CLS:FOR X=4 TO 127 STEP12
20 FOR Y=0 TO 47
30 SET(X,Y)
40 NEXT Y:NEXT X
50 FOR Y=2 TO 47 STEP6
60 FOR X=0 TO 127
70 SET(X,Y)
80 NEXT X:NEXT Y
100 GOTO 100
200 CLS:FOR X=0 TO 127 STEP8
210 FOR Y=0 TO 47 STEP6
220 SET(X,Y)
230 NEXT Y:NEXT X
250 GOTO 250
```



FLASH

BY

THEODORE C. HINES ROSANN COLLINS JERRY RUSSELL LINDA SPENCER

Dr. Hines is a professor in the Library Science/Educational Technology Division of the School of Education at the University of North Carolina at Greensboro, 27412; Jerry Russell, Rosann Collins, and Linda Spencer are Research Associates with the Children's Media Data Bank project at the University.

They welcome opportunities to work with other SOL owners in program exchanges. They can supply cassette tapes of Flash in Processor Technology CUTS format at $5 each to cover costs. Send inquiries to Dr. Hines.

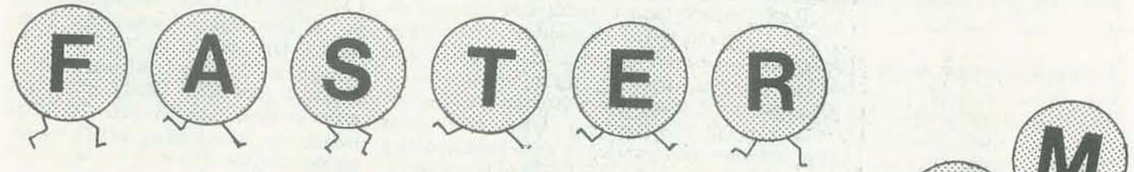
I just had a hot flash*!! I was looking at the fourth program response (P=4). Unless you are running this program in an all male school, you may wish to change that particular response to "Atta Person!!" – RZ*

Flash is a game of word recall. A word is *flashed* on the CRT screen for a short time, after which the player is asked to type the word on the keyboard. The program may be used for word recognition drill, spelling drill, reading instruction, or for that matter, just as a fun game. There are several variations of Flash: a teletype version for the Hewlett-Packard 2000F, for instance.

This Flash program is written in Processor Techonology Extended BASIC and is easily adaptable to other versions of BASIC. In this program, a session runs for 10 words, after which the player may choose to quit or continue. A cumulative score is kept. The player is offered a choice of word difficulty and length of time the word stays on the screen. Also, the player may set the number of seconds allowed for answering. Computer-generated responses to both correct and incorrect answers are varied to help maintain interest.

Words are stored in DATA statements and presented randomly from the appropriate group. An array keeps track of those words already used in a given session. By changing the words in the DATA statements the game is adaptable to many different levels.

```
LIST
 10 PRINT "&K"
 20 DIM L4$(65),L3$(65),W$(65),B$(65),C$(65),H$(65)
 30 DIM A(30)
 40 LET L4$="
 50 LET W$="FLASH": GOSUB 1490
 60 PRINT : PRINT
 70 LET W$="Copyright 1978 by Jerry Russell": GOSUB 1490
 75 LET W$="and Theodore C. Hines": GOSUB 1490
 80 LET W$="University of North Carolina at Greensboro":
       GOSUB 1490
 90 LET W$="Greensboro, North Carolina": GOSUB 1490
100 LET W$="1978": GOSUB 1490
110 PRINT
120 PRINT : PRINT : PRINT
130 PAUSE 40
140 PRINT "&K"
150 REM- INSTRUCTION SEQUENCE
160 PRINT "DO YOU NEED INSTRUCTIONS (YES OR NO)";
170 INPUT A$
180 IF A$="" THEN 160
190 IF A$(1,1)="Y" THEN 200 ELSE 370
200 PRINT "&K"
210 LET W$="INSTRUCTIONS": GOSUB 1490
220 PRINT : PRINT
230 PRINT "This is the game of FLASH."
240 PRINT "It tests your ability to recall words."
250 PRINT
260 PRINT "A word will appear on the screen and "
270 PRINT "you will be asked to type back the word."
280 PRINT
290 PRINT "You will be given 10 words and then asked"
300 PRINT "if you want more."
310 PRINT "When you are ready to start, hit 'RETURN'."
320 INPUT "",X$
330 PRINT "&K"
340 FOR G=1 TO 30
350   LET A(G)=0
360 NEXT G
365 REMFORMAT/QUESTION SEQUENCE
370 PRINT "WOULD YOU LIKE DIFFICULT, MEDIUM, OR EASY WORDS?"
380 PRINT "(TYPE YOUR CHOICE.)";
390 INPUT M$
400 IF M$="" THEN 370
410 IF M$(1,1)="D" THEN  LET R=1
420 IF M$(1,1)="M" THEN  LET R=2
430 IF M$(1,1)="E" THEN  LET R=3
440 IF R<1 OR R>3 THEN 370
450 PRINT : PRINT
460 PRINT "DO YOU WANT THE WORDS PRINTED FAST OR SLOW ?"
470 PRINT "(TYPE EITHER 'FAST' OR 'SLOW')"
480 INPUT W$
490 IF W$="" THEN 460
500 IF W$(1,1)="F" THEN  LET C7=25: LET Z2=1
510 IF W$(1,1)="S" THEN  LET C7=75: LET Z2=1
520 IF Z2<>1 THEN 460
530 PRINT : PRINT
540 PRINT "WOULD YOU LIKE TO BE TIMED?"
550 PRINT "(TYPE 'YES' OR 'NO')"
560 INPUT P$
570 IF P$="" THEN 540
580 IF P$(1,1)="Y" THEN 610
590 IF P$(1,1)="N" THEN  LET Y5=1
600 GOTO 640
610 PRINT "PLEASE TYPE THE NUMBER OF SECONDS YOU WANT"
620 PRINT "TO HAVE TO ANSWER EACH QUESTION."
630 INPUT V
640 PAUSE 15
650 PRINT "&K"
660 PRINT "HERE WE GO!!!"
670 PRINT : PRINT : GOTO 680
680 FOR I=1 TO 10
690   IF R=1 THEN  RESTORE 1270
700   IF R=2 THEN  RESTORE 1350
710   IF R=3 THEN  RESTORE 1420
720   REM- PICKS WORDS
730   LET G=INT(RND(0)*30)+1
740   IF A(G)=1 THEN 730
750   READ C$
760   LET K=K+1
770   IF K=G THEN 790
780   GOTO 750
790   LET A(G)=1
800   LET K=0
810   RESTORE
820   PRINT : PRINT : PRINT
830   PRINT "READY....."
840   PAUSE 15
850   PRINT "&K"
860   CURSOR 8,20
870   SET DS=C7
880   PRINT C$
890   SET DS=0
900   PRINT "&K"
910   LET L=(10*V)
920   PRINT "WHAT WAS THE WORD ";
930   REM- INPUT TIMING
940   INPUT (0,L)B$
950   REM- CHECKING AND SCORING SEQUENCE
960   LET P=INT(RND(0)*5)+1
970   PRINT : PRINT : PRINT
980   LET H$=C$
990   IF Y5=1 THEN 1020
1000  IF B$="" THEN  PRINT "TIME'S UP....."
1010  IF B$="" THEN  PRINT "THE WORD WAS....."!H$: GOTO 1050
1020  IF B$=C$ THEN 1070
1030  IF P<3 THEN  PRINT "NO, THE WORD IS...."!H$
1040  IF P>=3 THEN  PRINT "YOU MISSED IT. THE WORD WAS..."!H$
1050  LET U=U+1
1060  GOTO 1130
1070  LET Z=Z+1
1080  IF P=1 THEN  PRINT "YOU GOT IT!!!"
1090  IF P=2 THEN  PRINT "THAT'S IT !!!"
1100  IF P=3 THEN  PRINT "GOOD !!"
1110  IF P=4 THEN  PRINT "ATTA BOY!!"
1120  IF P>=5 THEN  PRINT "THAT'S RIGHT!!!"
1130 NEXT I
1140 PRINT "&K"
1150 PAUSE 20
1155 REM- SCORING SEQUENCE
1160 PRINT "YOU GOT "!Z!" RIGHT AND "!U!" WRONG"
1170 PRINT "OUT OF A TOTAL OF"!Z+U
1180 PRINT : PRINT
1190 LET R=((Z/(Z+U))*100)
1200 PRINT "YOU GOT "!R!" PERCENT OF THE WORDS CORRECT"
1210 PRINT : PRINT
1220 PRINT "WOULD YOU LIKE MORE WORDS? (YES OR NO)";
1230 INPUT T$
1240 IF T$(1,1)="Y" THEN 330 ELSE 1260
1250 PRINT : PRINT
1260 PRINT "SEE YOU LATER!"
1265 REM- WORD BANK
1270 DATA "FACETIOUS","PARSIMONIOUS","PRECIPITATION","PHYLA"
1280 DATA "DIPSOMANIAC","INCREDULITY","SYMPHONIC","GRADUALLY"
1290 DATA "CARNIVOROUS","WHETHER","QUAGMIRE","UNGUENT"
1300 DATA "DELICIOUS","SUBSTANTIAL","CENTENNIAL","ASSUMPTION"
1310 DATA "FRAGMENTS","JUDICIOUS","VIVIDLY","GREGARIOUS"
1320 DATA "OPINION","HARRASS","OBSERVANT","ENTITY","QUIPS"
1330 DATA "INDEFENSIBLE","PREDESTINATION","MATRIX"
1340 DATA "QUALITY","REHYDRATED"
1350 DATA "PRACTICE","WITHHOLD","QUESTION","DIVIDED"
1360 DATA "GRACIOUS","FRAMED","SPECIAL","HAPPINESS"
1370 DATA "ARRANGE","JUMPED","QUICK","FLATTEN","COSTLY"
1380 DATA "BUSINESS","MULE","NUMBER","PRODUCE","DISTANT"
1390 DATA "KICKED","LOWER","HIGHER","SENSE","EXACT"
1400 DATA "INSIDE","GRABBED","ZOO","UNSAFE","HARDLY"
1410 DATA "MOTION","FIXED"
1420 DATA "FACE","DOG","HILL","YELL","RED","GREEN"
1430 DATA "SIDE","TEAM","SLIDE","FUNNY","PLAY"
1440 DATA "KICK","RACE","DOWN","FIX","PATTED"
1450 DATA "WITH","FLOWER","OPEN","CLOSED","SMILE","TOP"
1460 DATA "GRASS","JUMP","ALL","HOUSE","ROAD","BLUE"
1470 DATA "ZOO","TRIP"
1480 END
1485 REM- CENTERING
1490 LET L2=INT((50-LEN(W$))/2)
1500 LET L3$=L4$(1,L2)
1510 PRINT L3$!W$: RETURN
```

BY PETER A. STARK

Peter Stark, from Mt. Kisco, NY, sends us a modified version of the Jumble routine that appeared in PC *(Vol. 7, No. 2, page 48) Sept.-Oct. 1978. His program runs on the SWTPC. Peter states that his version is faster than the original. Try them both and see . . .–RZ*

```
D
 0020 REM JUMBLE
 0030 REM BY M.C. HOFHEINZ
 0040 REM FROM PEOPLE'S COMPUTERS V7N2P48
 0045 REM MODIFIED BY P. STARK TO SPEED IT UP
 0050 REM ASSIGN A K$(Y) VALUE TO EACH LETTER
 0100 PRINT" TYPE IN A WORD OF 6 LETTERS OR LESS."
 0150 INPUT N$
 0160  L=LEN(N$)
 0200 FOR Y=1 TO L
 0300  K$(Y)=MID$(N$,Y,1)
 0400 NEXT Y
 0500 REM START NESTED LOOPS TO INTERCHANGE LETTERS
 0600 FOR A=1 TO L
 0620 FOR I=1 TO L
 0630  Q$(I)=K$(I)
 0640 NEXT I
 0650 REM TAKE OUT A-TH LETTER
 0655  P$(1)=Q$(A)
 0660  Q$(A)=Q$(L)
 0700 FOR B=1 TO L-1
 0701  M=L-1
 0702 FOR I=1 TO M
 0703  R$(I)=Q$(I)
 0704 NEXT I
 0705  P$(2)=R$(B)
 0710  R$(B)=R$(M)
 0715 IF L<3 THEN 1500
 0800 FOR C=1 TO L-2
 0805  M=L-2
 0810 FOR I=1 TO M
 0820  S$(I)=R$(I)
 0830 NEXT I
 0840  P$(3)=S$(C)
 0850  S$(C)=S$(M)
 0860 IF L<4 THEN 1500
 0900 FOR D=1 TO L-3
 0910  T$(1)=S$(1)
 0920  T$(2)=S$(2)
 0930  T$(3)=S$(3)
 0940  P$(4)=T$(D)
 0950  T$(D)=T$(L-3)
 0960 IF L<5 THEN 1500
 1000 FOR E=1 TO L-4
 1040  P$(5)=T$(E)
 1050  P$(6)=T$(3+E)
 1500 FOR I=1 TO L
 1600 PRINT P$(I);
 1700 NEXT I
 1800 PRINT" ";
 2005 IF L<3 THEN 2300
 2010 IF L<4 THEN 2200
 2020 IF L<5 THEN 2100
 2025 PRINT" ";
 2030 IF L<6 THEN 2070
 2070 NEXT E
 2100 NEXT D
 2200 NEXT C
 2300 NEXT B
 2400 NEXT A
```

Golf Handicapping

BY MILAN D. CHEPKO

Even though Milan has assured us that he is not a swinger, he has still produced a solid hit with the following program. I can see a network of TRS-80 computers eventually handling all of the functions of business and government within the city of Thief River Falls, MN. Grocery stores wired to banks; computer reminders of parking tickets; a voice-synthesized message when you phone Milan, saying, "The doctor is debugging." There are rumors that the whole system will involve six TRS-80 computers with a dual floppy disk each. Now, Milan, you said you didn't play golf. Why are you getting teed off? . . . —RZ

The hobby literature is full of programs that use the computer to simulate an opponent in a game, but few programs are available that mimic the role of larger computers – gathering, manipulating, and storing data. Most businessmen are impressed by the graphic and game capabilities of microcomputers, but they are usually more interested in practical uses. The following program was written primarily to solve a data processing problem for the local golf club. It demonstrates some uses the micro might have for local businessmen, who either use hand calculators or subscribe to large data-processing services.

I don't play golf, but in speaking with some of the club members, I found that members pay $3 a year cash for a periodic printout of their handicaps. They receive 10 listings a season, or 30¢/listing/member. The handicap is calculated by taking the average of the 10 lowest scores from the last 20 rounds; the US Golf Association course rating is subtracted from this average, and the result is multiplied by 0.96 (don't ask me why!). The final result, rounded, is the handicap.

The manipulation of the data is quite straightforward. The collection and storage of data proved to be tricky, since I don't yet have a disk. After trying several storage formats, I decided to always write out 20 scores, even if the member hasn't posted that many games. In this way, each member's data is represented on the tape in one continuous burst containing the name, number of valid scores (up to 20), the actual scores, and the handicap. When new scores are added, old scores are deleted according to age, and a new handicap is calculated. As presently written, the program will not give a handicap for less than 20 scores, although this could easily be changed if needed.

To make the service more attractive to the club, I increased the printout to include the member's scores so he can check them against his own records. I use the screen printer and it doesn't really matter how much material is on the screen! I also included an automatic printing option that activates the printer when the display is completed. This can be taken out, but I've found it better to waste a few feet of paper than to forget to hit the "print" switch. When you have made as many copies of the display as needed, just hit any key (except "BREAK") to continue the program.

Most details of the program are obvious from the flow chart and margin notes. It seems to take about 5K of memory with the arrays. You could probably run it in a 4K machine by deleting the cumulative listing (starting at line 2000). Also, it could be rewritten to run in Level I by assigning different portions of array A ( ) to replace B( ), and deleting lines 2000 on. It goes without saying that high-quality tape should be used for data storage, and making a second copy is always a good idea. Finally, if the program crashes, type "GOTO 2000" to recover the master listing before restarting the program.

Since the paper for the screen printer costs about a penny a foot when ordered directly from Nicolet Paper, DePere, Wisconsin 54115 (minimum order is a case of 24-200 foot rolls at $50.40), and each printout uses less than a foot of paper, it should be possible to offer this service at considerably less than the current 30¢/listing they are paying. In doing some trial runs, I found that it takes between one and two minutes to enter, update, print, and store each member's entry; at that rate, it would take about two hours to do 100 members!

SUMMARY OF VARIABLES

A$	name of current player	H	current handicap
B$	date	I, J, K, L	counters
D$	used to terminate display	M	number of players accumulated in arrays
A( )	player's current scores (A(1)=oldest)	Q	autoprint flag
B( )	bubble-sorted scores	Y	USGA course rating
P$( )	array of players' names	Z	average of 10 lowest scores
S( )	array of players' handicaps	N	number of valid scores . . . entered from tape or accumulated by adding new scores

```
01 '*** MILAN D. CHEPKO, M.D.
2 '*** THIEF RIVER FALLS, MN      56701
3 '*** 23 DEC 78
10 CLS:PRINT"*** G O L F   H A N D I C A P ***
20 CLEAR1000:DEFINT A-X
30 DIM A(40),B(40),S(40),P$(40):M=0
40 FOR I=1TO40:A(I)=0:B(I)=0:S(I)=0:P$(I)=" ":NEXT I
50 PRINT:INPUT"WHAT IS THE DATE";B$
60 PRINT:INPUT"ENTER '0' FOR MANUAL OR '1' FOR AUTOPRINT";Q
70 PRINT:INPUT"WHAT IS USGA COURSE RATING";Y
99 '*** MODE TABLE ***
100 CLS
110 PRINT"SELECT APPROPRIATE MODE FROM THE TABLE."
120 PRINT"    1=ENTER OLD DATA FROM TAPE"
130 PRINT"    2=RECORD UPDATED DATA ON TAPE"
140 PRINT"    3=ESTABLISH FILE FOR NEW MEMBER"
150 PRINT"    4=DISPLAY CUMULATIVE LIST"
160 INPUT I:IF I>4 GOTO100
170 ON I GOTO 200,500,400,2000
199 '*** LOAD OLD DATA TAPE ***
200 CLS:FOR I=1TO40:A(I)=0:B(I)=0:NEXT I
210 PRINT:PRINT"INSERT OLD DATA TAPE."
211 PRINT:PRINT"CHECK CONNECTIONS AND VOLUME SETTING."
212 PRINT:PRINT"PUSH 'PLAY' BUTTON ON RECORDER."
213 PRINT:INPUT"HIT 'ENTER' WHEN READY";D$
214 INPUT#-1,A$,N,A(1),A(2),A(3),A(4),A(5),A(6),A(7),A(8),A(9),A(10),A(11),A(12)
    ,A(13),A(14),A(15),A(16),A(17),A(18),A(19),A(20),H
220 CLS:PRINTA$,,"OLD SCORES="
225 FOR I=1TO10:PRINTA(I);" ";:NEXT I:PRINT
226 FOR I=11TO20:PRINTA(I);" ";:NEXT I
230 PRINT:PRINT"PREVIOUS HANDICAP=";H
240 GOSUB1000
250 IF A(I)=1000 THEN A(I)=0:GOTO220
260 I=I-1:GOSUB1100
270 IF I=20 THEN GOTO300
280 GOSUB1300
300 IF Q=1 THEN OUT 254,255
310 D$=INKEY$:IF D$="" GOTO310
320 M=M+1:P$(M)=A$:S(M)=H:IF M=30 GOTO2000
330 CLS:PRINT"BE SURE TO SAVE THIS DATA ON TAPE!!!
340 PRINT:GOTO110
399 '*** ROUTINE TO ADD A NEW MEMBER ***
400 CLS:N=0:H=0
410 FOR I=1TO40:A(I)=0:B(I)=0:NEXT I
420 INPUT"INPUT PLAYER'S NAME (JOHN Q. SMITH)";A$
430 GOTO240
499 '*** ROUTINE TO SAVE DATA ON NEW TAPE ***
500 CLS:PRINT"PLACE NEW DATA TAPE IN RECORDER."
510 PRINT:PRINT"CHECK ALL CONNECTIONS."
520 PRINT:PRINT"PRESS THE 'RECORD' AND 'PLAY' BUTTONS."
530 PRINT:INPUT"HIT 'ENTER' WHEN READY";D$
540 PRINT#-1,A$,N,A(1),A(2),A(3),A(4),A(5),A(6),A(7),A(8),A(9),A(10),A(11),A(12)
    ,A(13),A(14),A(15),A(16),A(17),A(18),A(19),A(20),H
550 CLS:PRINT"REMOVE AND STORE THE DATA TAPE."
560 PRINT:PRINT"(CONSIDER MAKING A BACK-UP TAPE!!!)
570 PRINT:GOTO110
999 '*** ROUTINE TO ENTER NEW SCORES ***
1000 I=20:PRINT:PRINT"YOU MAY INPUT UP TO 19 NEW ENTRIES."
1010 PRINT"TERMINATE THE LIST BY ENTERING '0'."
1020 PRINT"TO CORRECT ERRORS, ENTER '1000' AND START OVER."
1030 I=I+1:INPUT"NEXT SCORE";A(I)
1040 IF A(I)=0 OR A(I)=1000 THEN CLS:RETURN
1050 GOTO1030
1099 '*** ROUTINE TO PRINT OUT MEMBER'S STANDING ***
1100 CLS:PRINTTAB(15)"THIEF RIVER GOLF CLUB"
1110 PRINTTAB(10)"GOLF HANDICAP FOR USGA RATING=";Y
1140 PRINTA$;TAB(40)B$:PRINT
1150 PRINT"HERE ARE YOUR PREVIOUS 20 MOST RECENT SCORES"
1160 FOR J=1TO10:PRINTA(J);" ";:NEXT J:PRINT
1165 FOR J=11TO20:PRINTA(J);" ";:NEXT J:PRINT
1170 IF H=0 PRINT"NO HANDICAP AVAILABLE":GOTO1190
1180 PRINT"YOUR HANDICAP WAS";H
1190 IF I=20 PRINT"NO NEW ENTRIES AVAILABLE":RETURN
1200 PRINT"YOUR LATEST SCORES ARE"
1210 FOR J=21TO I:PRINT A(J);" ";:NEXT J:PRINT
1220 RETURN
1299 '*** COMPRESSES SCORES ***
1300 IF N=20 GOTO1340
1310 FOR J=21TO I:N=N+1:A(N)=A(J):NEXT J
1320 IF N<20 THEN H=0:PRINT"NO HANDICAP AVAILABLE":RETURN
1330 I=N:N=20:IF I=20 GOTO1380
1340 FOR J=21TO I
1350 FOR K=1 TO I
1360 A(K-1)=A(K)
1370 NEXT K:NEXT J
1380 FOR I=1TO20:B(I)=A(I):NEXT I
1399 '*** SORTS SCORES, CALCULATES HANDICAP ***
1400 J=0
1410 FOR I=1TO19
1420 IF B(I)<=B(I+1) GOTO1440
1430 K=B(I):B(I)=B(I+1):B(I+1)=K:J=1
1440 NEXT I
1450 IF J=1 GOTO1400
1460 Z=0:PRINT"YOUR 10 LOWEST SCORES WERE"
1470 FOR I=1TO10:Z=Z+B(I):PRINTB(I);" ";:NEXT I:PRINT
1480 Z=(Z/10)-Y:IF Z<=0 THEN H=0:GOTO1500
1490 H=INT(.96*Z)
1500 PRINT:PRINT"YOUR NEW HANDICAP =";H:RETURN
1999 '*** PRINTS OUT LIST OF UP TO 30 MEMBERS ***
2000 CLS:FOR I=1TO29 STEP2
2010 PRINT S(I);TAB(4)P$(I),"
2020 PRINTTAB(32) S(I+1);TAB(36)P$(I+1):NEXT I
2030 IF M<30 GOTO2060
2040 M=0:OUT 254,255
2050 FOR I=1TO30:P$(I)="":S(I)=0:NEXT I
2060 D$=INKEY$:IF D$="" GOTO2060
2070 CLS:PRINT"REMEMBER TO STORE DATA IF NECESSARY!!!
2080 PRINT:GOTO110
```

HUNT A New Class of Game

BY MICHAEL RICHTER

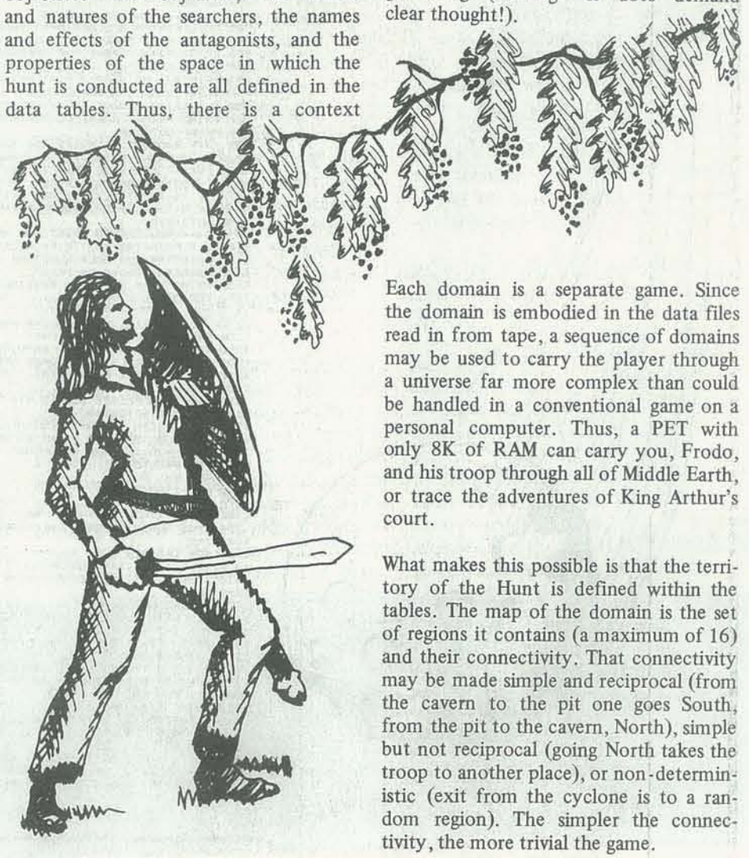
Why not a game where you define the game? That is the question Michael proposed to himself and implemented for the PET. You don't have to know how to program. You don't have to have a totally clear idea of the game you wish to play. You can invent a version, discover flaws, correct, improve, re-invent; do all the interesting parts of the process and leave the programming behind. – *RZ*

The use of games to exercise and occupy computers is virtually as old as computing itself. Most such computer games are variants of conventional games. Frequently, the principal difference is that the computer provides the opponent, so that a two-player game may be played alone.

The question may be asked: is there a meaningful class of computer game which has no conventional analogue? With Hunt, a prototype of such a concept is now available for the Commodore PET. It is written in Microsoft BASIC, so should be transferrable to other systems with little difficulty.

The concept underlying Hunt is that of a table-driven game. On the simplest level, the game is played with a set of data tables in the style of Quest, Dungeons and Dragons, Adventure, or any of a wide variety of both board and computer games. On the next level, that of a meta-game (in the sense of metaphysics or metamathematics), there is no conventional equivalent. Here, the player is defining the rules of the game, not merely playing it.

That definition is supported by an interactive program with substantial freedom. In Hunt, the context is established by the game-playing program and its associated interactive Huntwriter. The context is that of a search for a defined objective. The objective, the names and natures of the searchers, the names and effects of the antagonists, and the properties of the space in which the hunt is conducted are all defined in the data tables. Thus, there is a context (Hunt) which may have a virtually infinite variety of domains (specific games) at the discretion of the game's author. And that authorship does not require knowledge of any computer language or adherence to the formalisms of programming (although it does demand clear thought!).



Each domain is a separate game. Since the domain is embodied in the data files read in from tape, a sequence of domains may be used to carry the player through a universe far more complex than could be handled in a conventional game on a personal computer. Thus, a PET with only 8K of RAM can carry you, Frodo, and his troop through all of Middle Earth, or trace the adventures of King Arthur's court.

What makes this possible is that the territory of the Hunt is defined within the tables. The map of the domain is the set of regions it contains (a maximum of 16) and their connectivity. That connectivity may be made simple and reciprocal (from the cavern to the pit one goes South, from the pit to the cavern, North), simple but not reciprocal (going North takes the troop to another place), or non-deterministic (exit from the cyclone is to a random region). The simpler the connectivity, the more trivial the game.

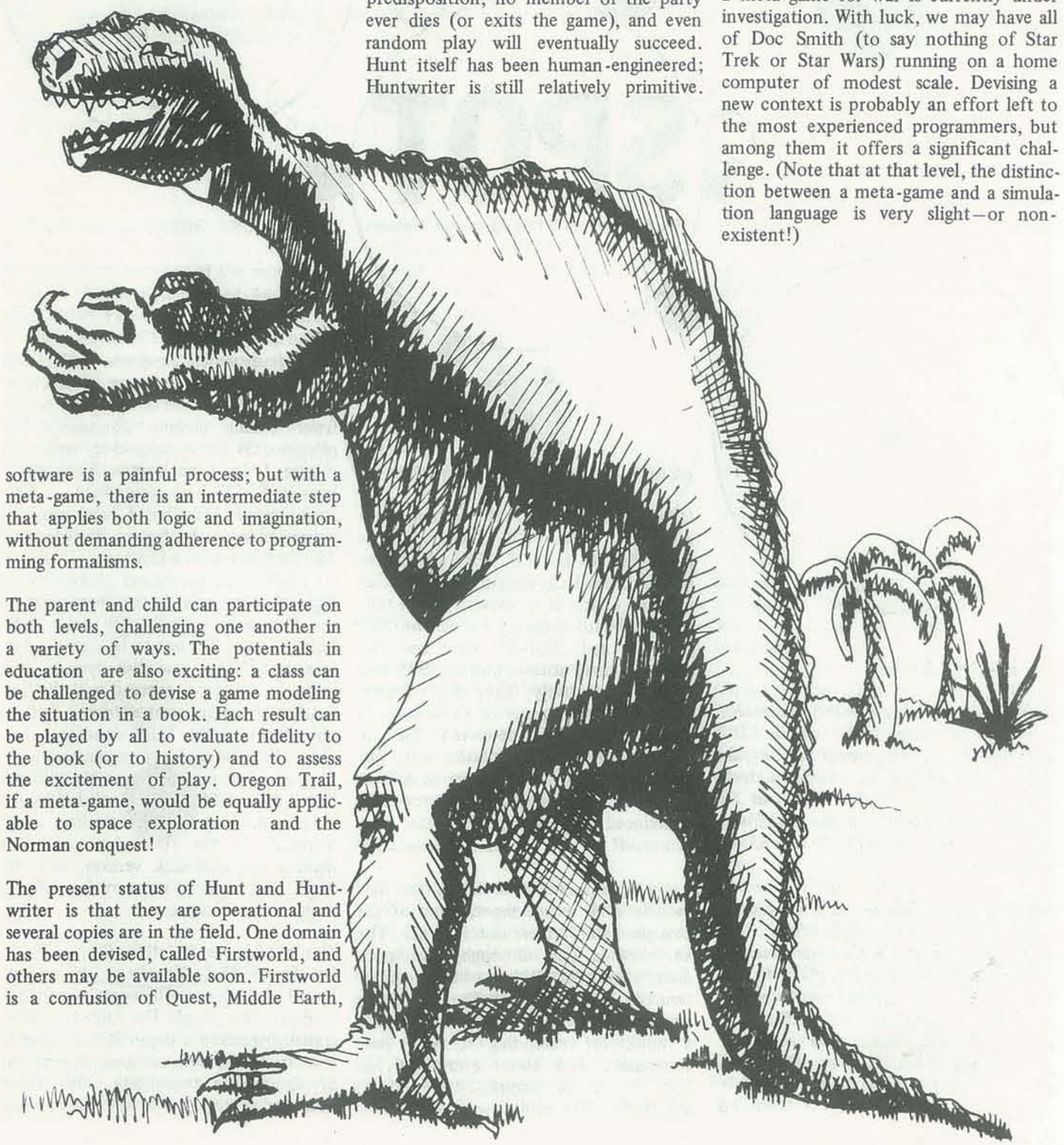
Philosophically, the concept of a meta-game offers several advantages beyond merely fitting a large problem into a small machine. Most important to the author, a level of computer involvement intermediate between playing a game and writing one is now available. The gap between the inventor of a self-designed game and its player is immense. To use a game to involve a non-programmer in software is a painful process; but with a meta-game, there is an intermediate step that applies both logic and imagination, without demanding adherence to programming formalisms.

The parent and child can participate on both levels, challenging one another in a variety of ways. The potentials in education are also exciting: a class can be challenged to devise a game modeling the situation in a book. Each result can be played by all to evaluate fidelity to the book (or to history) and to assess the excitement of play. Oregon Trail, if a meta-game, would be equally applicable to space exploration and the Norman conquest!

The present status of Hunt and Huntwriter is that they are operational and several copies are in the field. One domain has been devised, called Firstworld, and others may be available soon. Firstworld is a confusion of Quest, Middle Earth, Oz, and other ideas. The objective is a treasure reachable with a simply connected route; success requires exit from the domain, and another simply connected route will get you out.

The complete game can be played by one who knows the domain in about five minutes; without a map, it may take 20 minutes. Because of the author's predisposition, no member of the party ever dies (or exits the game), and even random play will eventually succeed. Hunt itself has been human-engineered; Huntwriter is still relatively primitive. Copies of the programs and Firstworld are available from the author for experimentation. After further checkout, they will be released more generally. To get a copy, please send your address, return postage and a suitable cassette to: 2600 Colby Avenue, Los Angeles, CA 90064.

Finally, a third level of play is available but not implemented. The possibility of a meta-game for war is currently under investigation. With luck, we may have all of Doc Smith (to say nothing of Star Trek or Star Wars) running on a home computer of modest scale. Devising a new context is probably an effort left to the most experienced programmers, but among them it offers a significant challenge. (Note that at that level, the distinction between a meta-game and a simulation language is very slight—or non-existent!)



SPOT

The Society of PET Owners and Trainers

HEARD AROUND THE QUAYSIDE

BY HARRY SAAL

Commodore's PET is a factory-assembled personal computer based on a 6502 microprocessor. The original PET, model 2001-8, is a $795 system that includes a keyboard, cassette tape unit, built-in TV screen, some graphics, upper and lower case, extended 8K BASIC, and 8K of user memory.

SPOT is devoted to the host of applications—routine and wild—which PET users have found for their machines, as well as to the nitty-gritty of repairs and modifications. In other words, almost anything relating to the PET is fit material for this column. Just send Harry your questions and ideas c/o PCC. He'll give each of them his careful attention.
—LB

Have you heard there are now two *more* models of the PET? Commodore is producing 16- and 32K PETs which have the old, small keyboard, and built-in cassette. (There is no cassette in the full-size keyboard models.) Called the PET 2001-16 and 2001-32, they cost the same as their sister models, $995 and $1195, respectively. They don't appear to be described in any of Commodore's literature, and it is not clear if they are available from dealers nationwide, but I've seen them for sale here in Silicon Valley. Anyone sense that the previously announced models might be later than promised?

I did spend some time trying the new models at a recent show; some of the changes I like, others not so much. The keyboard is nice, although it certainly doesn't have the "feel" of high-quality terminal keyboards. The machine language monitor is in ROM, and even has a hook for extending it with new commands. Just about everything has been moved, so memory map makers, get ready. The screen writing is much faster (ever watch an Apple scroll?) due to a nice hardware change, which prevents programs that POKE the screen from getting glitches of snow. The phosphor is green instead of white, a change I don't appreciate. The known bugs in BASIC have been fixed . . . Certainly others lurk, but I hope none as catastrophic as the former limitation of 256 elements to an array.

Read the fine print in the description of the Commodore single-disk drive carefully. It turns out *not* to be a single-disk version of the dual-disk system, but rather a cut-down system, with limited function ("a fast cassette drive"). It does not plug into the IEEE connector, but onto the new memory expansion port. That means it is not compatible with the "old" 4- and 8K PETs. Don't hold your breath too long for this product to be shipped. All the effort has gone into making the dual-disk version, and the single-disk model is not very far along in design at this writing.

Words fail me in describing Commodore's *Pet User's Club Newsletter*, Volume I, Issue 1. As happens with most Commodore mailings, few people I've talked to have actually received a copy. But nothing is lost. It is *bad,* with no new information of significance technically. This initial issue concentrates on descriptions of new PET hardware and cassette software. The printing is poor, literally illegible on page after page. Commodore expects people to pay $15 a year for this. Apparently, if you don't, you get no future mailings. Someone should show the editors *Contact,* which is Apple's (free) newsletter to all owners. Save your money, or better yet, send it to Commodore Business Machines, Limited, 3370 Pharmacy Avenue, Agincourt, Ontario, Canada M1W 2K4, for a decent publication!

Commodore has relocated to a new, all-solar-powered building, at 3330 Scott, Santa Clara, CA 95050. May the sun shine in!

MIDPENINSULA USERS' GROUP

On the first Wednesday of the month, at 7 p.m. in the cafeteria of Ford-Aerospace Corp., 3939 Fabian Way, Palo Alto, CA, the Midpeninsula PET Users' Group holds forth. The meetings are open to all. There, in the atmosphere of an electronic oriental bazaar, gather neophytes and experts, shoppers and vendors. Lately, there have been about 150-200 people and 40 PETs in attendance.

You can take a look at disk systems, such as the Computhink, Commodore's, or the Nestar Cluster/One, which was simultaneously connected to 15 PETs at a recent meeting while loaded with hundreds of programs from the free user library. Two different models of keyboard were there, along with light pens sound generators, speech encoders and transformers, and a variety of printers. Marv Vanderkooi, club president, demonstrated his PET-to-R2D2 radio controlled interface, whereby you control the robot from the PET keyboard, and can later edit and play back the commands from tape. Numerous software packages are tried out, giving people a chance to evaluate before investing money on programs. The club is quite strict about copying commercial software, and members police things well—ever since a few stern lectures a while back. Each month there are new wonders on display, and more programs in the library. Write and tell me what your local club is like.

REVIEWS

There are no software reviews this time. I will be happy to do them in the future, but only if we get a copy of a program to try out. *Recreational Computing* gets lots of announcements of software, but it is impossible to judge quality and novelty without those tapes!

PET Cassette Exchange

Len Lindsay continues the fine job that he has done in producing the *PET Gazette,* his free and informative newsletter, by coming up with the *PET Cassette Exchange.* He has put together a tape with 6 entertaining games (Blackjack, Snake, Stars, Tommy Termite, Chase and Hangman) and offers free copies to any hospital that requests it, to help cheer up lonely patients. Have your local hospital's volunteer services director request the Children/Hospital Games tape, on letterhead, to get a copy. (Write *PET Gazette,* 1929 Northport Dr., Room 6, Madison, WI 53704.) Then take your PET to the hospital and get involved. (Yes, you can copy the programs for your own use, but first let the patients play!)

Cursor Magazine

Cursor Magazine, published by Ron Jeffries, P.O. Box 550, Goleta, CA 93017 continues to be the best buy around. I was pretty skeptical when I first heard about it. How does this guy manage to collect such a nice variety of high-quality programs every month? I still don't really know, but every month seems to bring still better and better programs.

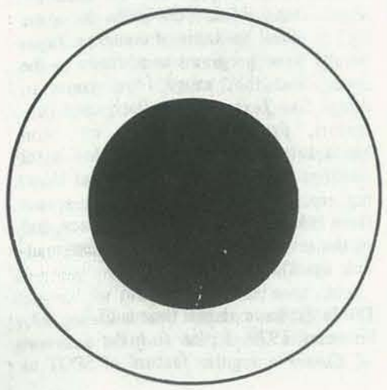
Cursor is distributed on cassette tape, with a one-page information sheet. The first program is always the "cover," a highly entertaining visual (or musical), which stops when you push the space bar, to reveal the table of contents. Tapes usually have programs in addition to the cover, and they range from games to things like text editors, flash card generators, project estimation, etc. Ron has a love of tools and includes useful routines to handle INPUT without blowing up, packing and unpacking spaces from BASIC programs (to save space, and in the reverse direction, make them readable again).

The latest issue at this time is *Cursor* #7, February 1979. I plan to make a review of *Cursor* a regular feature of SPOT as

a challenge to Ron to keep up the good work. February's cover is a kaleidoscopic pattern, and is pretty similar to others I've seen before. Pretty, but not very new. The *Pricer* program is useful for anyone preparing bids on almost any kind of job. You enter DATA statements which reflect wage rates. The program then prompts you for information about the job to be done, how long it will take, how many hours or months each person is putting in, overhead costs, any direct costs, etc. and then gives a detailed table of costs and profits.

Sound! is my favorite on this month's tape. It is a library of sound effects, about 20 of them, ranging from French police cars to flying saucers, and everything in between. Great job! *Mind* is a version of the board game Mastermind, and much better done than any previous PET version I've seen. Not that new, but it definitely replaces my old copy. *Fball* (Football) is also similar to other games I've seen, but with good graphics, a real time clock, complete with timesout, and a good competitive interaction with the PET, your opponent. Much better than other Footballs I've seen. Last on the tape is *Paper*, a cute "wallpaper" designer. Watch the screen for a good gag at the end.

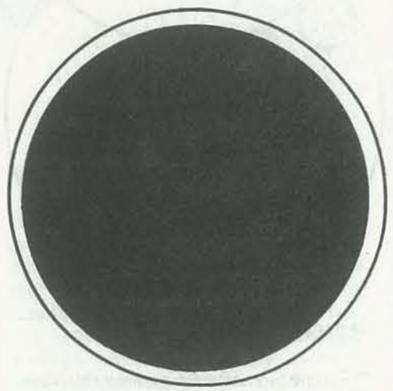
Is it worth it? *Cursor* now costs $36 a year, i.e. $3 a month. I think it is an incredible bargain. I've never had trouble loading a tape, thanks to Ron's very own high-speed duplication machine which he keeps in good shape. I am tempted to go back and tell you about the first six issues, but instead suggest you buy back copies for yourself.



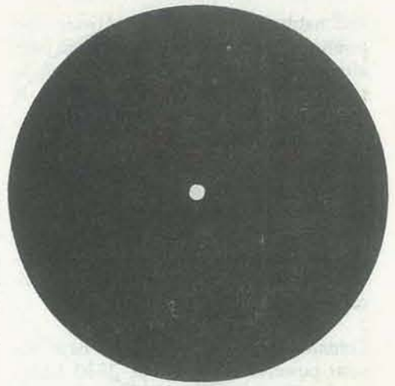
32 BASIC Programs for the PET Computer

This recent book, by Tom Rugg and Phil Feldman, is published by Dilithium Press, which describes it as "an absolute must for the PET owner." Well, it isn't. Some of the programs might be useful, or enjoyable, but unless you see something in it that you really *need*, you won't get much from the book. The programs don't exhibit any hint of expert programming style. My favorite is the beginning of *Vocab*, which reads . . .

```
300 GOSUB 1000
400 GOSUB 2000
500 GOSUB 3000
600 GOSUB 4000
700 GOSUB 5000
900 IF E=0 THEN 500.
```



This program and all the others are essentially devoid of comments, except for the title and copyright notice. Fear not. Who'd rip this stuff off? It is a shame the programs are not worthier, for the overall format of the book is quite laudable. Each program is written up with a section on program purpose, instructions, a sample run, suggestions for change, an overview of line numbers and what they do, and the variables used and their meaning, along with a set of suggested projects. This is fine, but none of the programs contain instructions for their use *in* the programs. What did you do the last time you encountered a program like that?



PETABLE, a workbook for PET

This sloppily done discussion of the PET hails from Richard Mansfield, P.O. Box 461, Philipsburg, PA 16866. For $4.95 you get 40 pages of some of the worst English I've read in years (typed in about the same quality), giving a rambling look at some BASIC for the PET. Save your money . . . At the end, there is an ad which says "If you have enjoyed *PETABLE*, you will love *Program*, our monthly magazine on cassette." What do I put for an ELSE after the IF?

CASCADES

The next page contains a program I wrote that is fun to watch. I think it makes good use of the PET graphics, and illustrates a number of techniques you should pick up. As things move around on the screen, it PEEKs into the screen buffer to see what's there, rather than keeping a large separate array of auxiliary information. Line 440 is funny to look at, until you realize that R and L are logical quantities, TRUE or FALSE (internally −1 or 0), that are used in the lines that follow.

What you should do with a program like this is *change it*. Try it a few times, and then make up your own idea of what it should do. Play with some of the parameters. Try bigger and smaller values for LNG on line 140, or MAX on line 130, for example. Set BOX=96 in line 130 and watch what happens. Make the boxes go vertically instead of horizontally. And especially, change the rules for what happens when the front of a drip gets stuck, from 600 on. Make it break, go up, and so on. Have fun!

CASCADES

```
100 REM CASCADES
110 REM COPYRIGHT 1979, HARRY J. SAAL
120 REM PERMISSION GRANTED FOR NON-COMMERCIAL USE
130 WALL=160: BALL=81: BOX=102: MAX=100
140 LNG= 6: DIM DRP(LNG)
150 SC=32768:PRINT"[CLEAR]"
160 REM MAKE THE WALLS
170 FOR I=0 TO 24
180 POKE SC+I*40,WALL
190 POKE SC+I*40+39,WALL
200 NEXT I
210 FOR I=0 TO 39
220 POKE SC+24*40+I,WALL:NEXT I
230 REM MAKE THE BARRIERS
240 FOR I=1 TO MAX
250 P=SC+INT(1000*RND(1))
260 FOR J=0 TO 3
270 IF PEEK(P+J)=32 THEN POKE P+J,BOX
280 NEXT J,I
290 FOR I=1 TO 38:POKE SC+I,32:NEXT I:REM MAKE A HOLE
300 REM DROP A NEW DRIP
310 S=SC+20: D=40 :CLNG = LNG
320 FOR I=1 TO CLNG:DRP(I)=0:NEXT I
330 DRP(CLNG)=S
340 IF PEEK(S)<>32 THEN 660
350 REM DRAW HEAD AND TRY TO MOVE
360 POKE S,BALL
370 REM TRY DOWN FIRST
380 IF PEEK(S+40)=32 THEN D=40: GOTO 510
390 NS=S+D: REM  NEW POSITION
400 IF PEEK(NS)=32 THEN 510
410 REM IF WE ARE GOING SIDEWAYS, GIVE UP
420 IF D<>40 THEN 600
430 REM CANT MOVE, CHECK LEFT AND RIGHT
440 R=32=PEEK(S+1): L=32=PEEK(S-1)
450 REM IF CAN GO BOTH WAYS, RANDOMLY CHOOSE
460 IF R AND L THEN D=1+2*(RND(1)<.5):GOTO 510
470 IF R THEN D=1:GOTO 510
480 IF L THEN D=-1:GOTO 510
490 GOTO 600: REM STUCK, SPLIT OFF THE HEAD AND TRY AGAIN
500 REM HERE WE MOVE BY D
510 TL=DRP(1): REM FIRST GET THE TAIL
520 IF TL<>0 THEN POKE TL,32: REM BLANK TAIL
530 REM ADVANCE ALL PARTS
540 FOR I=2 TO CLNG
550 DRP(I-1)=DRP(I)
560 NEXT I
570 DRP(CLNG)=S
580 S=S+D: DRP(CLNG)=S
590 GOTO 360
600 REM WE ARE STUCK, TRY THE NEXT ONE IN
610 IF CLNG<=2 THEN 310
620 CLNG=CLNG-1
630 S=DRP(CLNG)
640 IF S=0 THEN 660: REM DONE IF OFF SCREEN
650 D=40: GOTO 360: REM TRY TO GO DOWN
660 REM WE ARE FINISHED, START ALL OVER
670 RUN
```

Reviews

CHINA TRADESMAN: TAIPAN
Taipan TRS-80, Level II, 16K
Cybernautics
Box 40132
San Francisco, CA 94140
$9.98

Tired of space wars? Hammurabi getting a bit old? Looking for a new thrill on your micro? Well, have we got a game for you!! Taipan—a simulation of a seagoing trader set in China in the middle 1800s.

The game is so intriguing that I played it for over six hours when I first received a copy. You start the simulation in the home port of Hong Kong. You are in debt; you have nothing in the bank. (Sounds real enough.) You do own a ship that is capable of carrying 50 units of cargo. There is a warehouse in which you can store excess goods.

There is a sometimes friendly money-lender who is prone to giving Buddhistic lectures on debt paying. You can borrow money from him to help get you going. Using the money, you are given the opportunity to buy cargo for the ship. The trade goods are: general cargo, arms, silk and opium. At the beginning, because of your financial state, you must trade mostly in the first two items. As you acquire money, you can move up to the more costly goods. But, watch out!! As your cargo gets more expensive, you attract more pirates. Li Yuen, the chief pirate, can be found in Hong Kong upon occasion. He may ask for a donation to the temple. Refuse him at your own peril!

Once your ship is loaded, you set sail to one of several ports. There are storms, unspecified pirates and Li Yuen's pirates upon those treacherous waters. When you arrive in port, the prices for the goods you are carrying may be priced so low that you are forced to sail again. Meanwhile, your debts back in Hong Kong are accruing interest at an alarming rate. I won't give away any more of the game. There are many surprises in the program that are best discovered by playing it. I mean, is it possible that Li Yuen could actually show up on the open sea and chase away a band of pirates that are attacking you?

Several adults have been at my home when my children were playing the game. They would often ask what it was that seemed to have the children so engrossed. The kids would usually be in the middle of a hot debate over the merits of buying silk or general cargo, for example. I know that there are two answers to that question: 1) It's a game, or 2) it's a simulation of a historical scenario that teaches them about tradeoffs. I always use the second response because I know that 1) I'm talking to an adult, and 2) I know what their next response is. They always ask how can that be possible!

At that point, I say sit down and try it. Four hours later . . . but you know the rest.

Reviewed by Ramon M. Zamora.

TALE OF TWO TREKS
Startrek-80
Trek-78
TRS-80, Level II, 16K
Farrell Enterprises
P.O. Box 4392
Walnut Creek, CA 94596
$9.98 each.

Clyde Farrell has a two-Trek offering that will bring your TRS-80 alive with Enterprises, Klingons and Romulans.

Startrek-80 is an enhanced version of this classic simulation. There are short and long-range sensors, warp engines, photon torpedoes, phasors, an experimental death ray, a self-destruct option, damage control, supernovas, a complete galactic record and random events that cause or affect all of these items. As Clyde mentions in his program statement, there are events that occur so infrequently it may take you weeks to discover them.

The galaxy for the game is a set of 64 quadrants each divided into 64 sectors. The information on your "control screen" is well organized and enough is always visible so that the game can progress rapidly. If you are a Trek collector, this one is a must!

Trek-78 is an "animated" version of the game that utilizes the graphics of the TRS-80. The ships of the various parties are drawn on the screen; each one distinctive. When a photon torpedo is fired, you see it travel across space. The Romulans move about and are hidden by a cloaking device. They are invisible, until they attempt to fire a phasor. Sometimes they are friends; sometimes not. They sometimes fire on Klingons.

When phasors are being fired, each ship that is being hit lights up with the "halo" of the energy field. When you move to a new quadrant, you watch as the Enterprise builds up to warp speed and then winks out of sight at the edge of the quadrant that is being left. Here again, an old favorite is re-created in a new and exciting way.

Just a note concerning Clyde's tapes. I successfully load all of his tapes at a volume setting of between six and seven. His tapes are only for a Level II TRS-80 with 16K of memory.

Reviewed by Ramon M. Zamora.

TRS-80 TRON
CLOAD Magazine
Box 1267
Goleta, CA 93017
$36/year, $3.50/issue

At the Computer Faire in Los Angeles last November, it was stated that more than 175,000 TRS-80 computers had been sold so far and that approximately 12,000 units were being manufactured and sold by Radio Shack each month.

As a TRS-80, Level II—16K owner, I have purchased many publications to find programs to run on my machine. There certainly is no lack of material, as a glance at the advertising sections of hobby computer publications shows.

One major lack in these publications, however, is reviews that tell the TRS-80 owner which programs are good, bad, or indifferent. Obviously, most new computer owners can't purchase all that is offered. And there's nothing more frustrating than discovering a cassette program—for which you paid anywhere from $7.95 to $20—to be third-rate. Or worse, a recopy of a program already owned but with a changed title.

To take the first step in solving this problem for TRS-80 users, I am starting a series of reviews. This first review deals with the TRS-80 cassette magazine *CLOAD*, issue #9, November 1978. I will review more recent issues of *CLOAD* in future articles.

CLOAD is an excellent magazine, published on cassette with a short newsletter included. Most programs are listed twice, in both Level I and II, and can be loaded with 4K, though there are some outstanding programs listed in Level II-16K which are not listed for Level I. If a tape will not load, *CLOAD* will send you another copy upon return of your defective cassette.

The price of $3.50 (add 6% sales tax in California) for a single cassette—or $36 for a one-year subscription—is probably the lowest price going for a series of software programs. Normally, there are five programs in Level II and four in Level I. There is an animated "cover," which is also a program. So the cost works out to approximately 60 cents per program.

CLOAD was experiencing some problems in duplicating their tapes. But publisher Ralph McElory says that the problem has been licked, and *CLOAD* will be back on schedule shortly.

CLOAD is written by a number of programmers. There are six programs in this issue of *CLOAD: CLOAD* Cover, Instructions for Artillery, Artillery, Ohm's Law, Cat and Mouse, and Crushman.

The cover is visually interesting and an example of what the "new media" is likely to become. Artillery is slightly interesting and similar to formerly published games in *Kilobaud*. The main difference is the graphics and the use of side remarks. Ohm's Law is a good CAI (Computer Assisted Instruction) program for anyone just learning electronics. It is of minor interest if you are already proficient. Cat and Mouse has some problems and is not one of the better games *CLOAD* has published. Crushman is very similar to Hangman except there is an excellent graphic representation of an android within a box. This android will be crushed and buried if you make nine errors in guessing a random word. The words are fairly difficult, though you do have the choice of five levels of play. You can also change the words or enter new data if you desire. The graphics are very fast and many POKE statements are used. This is an excellent program and well worth the $3.50 cost of the cassette. It will assist beginning programmers in using the POKE statements to list fast graphics on the TRS-80.

Reviewed by James F. Fouke
El Granada, CA

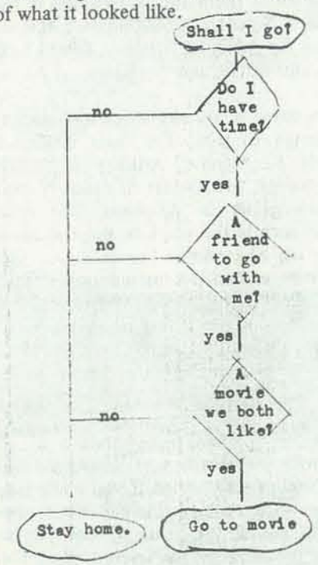
Next time we will look at CLOAD *# 10, December 1978, plus other TRS-80 material.* CLOAD *# 10 has two programs, a four-color map problem and a name-the-states drill, that kids (and even teachers) would like.* — *RZ*

COMPUTERS: A FIRST BOOK
By Linda O'Brien
Franklin Watts, 1978
68 pp., $4.90.

BOOK REPORT

I have just finished a book on computers by Linda O'Brien. The book was called *Computers*. The book tells about the history of the computers and how computers are made. I didn't understand very much of it. I think it is too old for

me. I am almost 9 years old. Some of it that I did understand, I liked. I liked the part where it told how the computer thinks. The computer solves a problem by asking itself yes/no questions. There was a diagram about movies. This is sort of what it looked like.



I think thes first part should be "will my mother let me?"

Reviewed by Suzanne Hofland, age 9
Oak Knoll School, Menlo Park, CA

BRING IN THE ORCS!
Lord of the Rings: The Movie

Last issue we brought you three reviews of the movie "Lord of the Rings" – *each written by a serious student of the literary trilogy. This time we present the reaction of a total Tolkien innocent. Susan Payette didn't know Gandalf from Gollum when she walked in the theater but, well, let her tell it . . .* *–LB*

A friend of mine had a cat named Hobbit. It was the ugliest cat I had ever seen; a huge cat with large feet that had wild hairs sticking out in every direction. Having never read the Tolkien books, I never really appreciated that cat . . .

My appreciation finally came as a result of going to see *Lord of the Rings* with my friend Bob. I had protested that I probably wouldn't understand a thing about the movie. For I was one of those few people who had managed never to buy, read, or analyze a Tolkien book. In my generation, it's akin to saying you never used Clearasil or watched "American Bandstand."

But, with reassurances that he would provide a running narrative as good as Howard Cosell, Bob and I went to see the movie.

As soon as we entered the popcorn-aroma lobby, I had a *deja-vu* feeling and could relate the experience to other times: going to a new school where everyone knew the location of the bathroom but you; singing dirty songs at Girl Scout camp and not knowing the words, but singing anyway . . . and so on. As we wove our way through the Tolkien aficionados in the lobby, I decided that even if I didn't understand the movie, it would help to rely on my usual fortifications of ice cream bon-bons and Pepsi. I had my first preview of what I had gotten into when I had to ask Bob who the person on the poster was that looked like a windblown Merlin. Bob patiently explained it was Gandalf the Wizard who would help Frodo the Hobbit.

The lights dimmed; I stuffed my cheeks with ice cream and we were off! The animation in the introduction was so realistic I found it hard to believe they were not using real people. Bob had done his homework and provided me with instant answers to all of my instant questions. Luckily, we went on an off night when 30, rather than 300, loyal Hobbiters were in the theater with us. I appreciated that when I first saw the Hobbits and had to ask, "Why do they have such big feet? Hairy feet?" Bob replied serenely, "Because they are Hobbits." I soon learned that this pronouncement was sort of a "what's-so" philosophy about Tolkien. I soon fell in love with Frodo (hairy feet and all), though I had trouble understanding why Frodo was quivering about the ring. I figured that if he put on the ring, all would be well. It wasn't until he finally did put the ring on that I realized its significance.

I didn't always understand what was going on, but I did start stereotyping the characters. I had Strider (Aragorn, son of Arathorn) as the Clint Eastwood of the bunch (and not bad at that!) I was glad to see that they included a Viking just for good measure, and of course a Fairy (Legolas).

I bit my knuckles along with everyone else when the orcs captured Merry and Pippin. When they were running to keep up with the orcs, I gently slid up to Bob and whispered, "See, if they didn't have those big hairy feet, they could keep up."

At times, the voices did not match the image I was seeing. This was especially apparent with Galadriel. She had such a beautiful voice and yet the homeliest eyes I have ever seen.

By the time we got to the battle at Helms Deep, I was hooked. I was on the edge of my seat as the battle started, cheering with the best of them. Especially for Strider (my Eastwood lookalike), hoping he wouldn't die – fantasizing about how I would revive him if he were other than animated! I relaxed as the battle was won. But then came that disturbing scene of my little friend, Frodo, sailing towards that sinister City of Doom. And was it my imagination, or was that ring getting heavier? The movie ended so suddenly, I felt incomplete.

Would Frodo live? Why was Gandalf wearing white? What was in the City of Doom? The whole movie suddenly seemed like a mirage in the desert of my mind.

Threading our way through the theatre lobby to leave, I had a sudden "ah-ha." Now I understood that Berkeley bumper sticker: "Frodo Lives!"

Reviewed by Susan Payette
Mill Valley, CA

DRAGONSMOKE

BY THE DRAGON & FRIENDS

FANTASY & SCIENCE FICTION GAMES (and where to get 'em)

Archive Miniatures
1111 S. Railroad Ave.
San Mateo, CA 94402

Star Rovers (in preparation)

•

Fantasy Games Unlimited, Inc. (FGU)
P. O. Box 182
Roslyn, NY 11576

Chivalry and Sorcery (C&S)
Bunnies and Burrows
Flash Gordon & the Warriors of Mongo
Starships and Spacemen

•

Gamescience
Lou Zocchi & Associates
7604 Newton Dr.
Biloxi, MS 39532

Knights of the Round Table
Space Patrol
Superhero 2044

•

TSR Hobbies, Inc. (TSR)
P. O. Box 756
Lake Geneva, WI 53147

Dungeons and Dragons (D&D)
Gamma World
Empire of the Petal Throne (EPT)
Metamorphosis Alpha (MA)
Star Probe
Star Empires

•

Tyr Gamemakers Ltd.
P. O. Box 414
Arlington, VA 22210

Space Quest
Bushido

•

Metagaming
P. O. Box 15346
Austin, TX 78761

Monsters! Monsters! (M!M!)

•

The CHAOSium
P. O. Box 6302
Albany, CA 94706

RuneQuest (RQ)

•

Flying Buffalo, Inc.
P. O. Box 1467
Scottsdale, AZ 85252

Tunnels and Trolls (T&T)
Starfaring

•

Game Designers Workshop (GDW)
203 North St.
Normal, IL 61761

En Garde!
Traveller

•

Legacy Press
217 Harmon Rd.
Camden, MI 49232

Legacy

•

James E. Mathis
2428 Ellsworth (#102)
Berkeley, CA 94704

Arduin Grimoire
Welcome to Skull Tower
Runes of Death

•

Heritage Models, Inc.
9840 Monroe Dr. (Bldg. 106)
Dallas, TX 75220

Star Trek

•

MAGAZINES

ALARUMS AND EXCURSIONS (A&E)
Lee Gold
3965 Alla Rd.
Los Angeles, CA 90066

THE DRAGON (TD)
TSR Periodicals, Inc.
P. O. Box 110
Lake Geneva, WI 53147

THE SPACE GAMER (TSG)
Metagaming
P. O. Box 15346
Austin, TX 78761

SORCERER'S APPRENTICE
Flying Buffalo, Inc.
P. O. Box 1467
Scottsdale, AZ 85252

THE JUDGES GUILD JOURNAL (JGJ) and THE DUNGEONEER
Judges Guild
1165 N. University
Decatur, IL 62526

THE LORDS OF CHAOS (LOC)
Nicolai Shapero
8885 Earhart Ave.
Westchester, CA 90045

THE WHITE DWARF (WD)
Games Workshop
One Dalling Rd.
Hammersmith, London W6 0JD
England

THE WILD HUNT (TWH)
Mark Swanson
71 Beacon St.
Arlington, MA 02174

WARGAMING
Fantasy Games Unlimited, Inc.
P. O. Box 182
Roslyn, NY 11576

The above information was taken from the premier issue of a great new FRP magazine, Different Worlds, *published by the CHAOSium.*

HOBBY STORES

Our first list of hobby stores where you can see, touch and buy fantasy & science fiction games.

The Compleat Strategist
11 East 33rd St.
New York, NY 10016

CS&D
731 S. University Blvd.
Denver, CO 80209

Coulter Bennett, Ltd.
12158 Hamlin St.
North Hollywood, CA 91606

Gamemasters Hobbies
4627 Geary Blvd.
San Francisco, CA 94118

Games People Play
1105 Massachusetts Ave.
Cambridge, MA 02138

Lincoln Park Chess 'N Games
2526 N. Lincoln Ave.
Chicago, IL 60614

Nan's Toys & Games
5015 Westheimer
Houston, TX 77056

Outpost Hobbies
224 California Dr.
Burlingame, CA 94010

San Antonio Hobby Shop
2550 W. El Camino
Mountain View, CA 94040

Send us some more!

Announcements

Hardware

Apple Clock. This calendar/clock for the Apple II keeps time and date in 1ms increments for more than a year. Calendar, clock, and event timer functions are easily accessed from BASIC using routines carried in on-board ROM. Applications include programming a morning printout of appointments, timing events, and creating games in which elapsed time is important. It can be combined with Mountain Hardware's Introl Remote Control System for real-time control and monitoring of remote devices. Assembled and tested, $199. Mountain Hardware, Inc., 300 Harvey West Blvd., Santa Cruz, CA 95060. (408) 429-8600.

Vowel Power. Texas Instruments has released Vowel Power, the first in a series of plug-in modules for the Speak & Spell learning aid. Vowel Power expands the built-in vocabulary of Speak & Spell, and all S&S activities can be played with it. The accompanying book contains a number of engaging vowel-sound games. According to TI, the module offers a unique approach to mastering these sounds by presenting 140 words divided into four categories. Through hearing, spelling, and reading these particular words, says TI, a child can become familiar with the patterns that vowel sounds follow in English. Available for approximately $15 at retailers carrying Speak & Spell. For further information: Texas Instruments, Consumer Relations, P.O. Box 53, Lubbock, TX 79408.

Cluster/One. With this new hardware-software system, up to 15 microcomputers can use the same programs and data files simultaneously. The central unit, which contains disk drives and a controller, connects the individual computers via a high-speed parallel data bus. PET, Apple II, and TRS-80 computers can all be used with Cluster/One—in any combination. It is a low-cost alternative to traditional time-sharing systems—and makes real-time applications feasible.

According to its designers, Cluster/One "is ideal for use in the classroom, laboratory, or program development environment. In the classroom, a Cluster/One system lets everyone get down to work in a matter of seconds. Programs are loaded a hundred times faster and much more reliably than from tape cassette. Students can be working on similar or identical programs, or doing independent work."

For more information, contact Nestar Systems, Inc., 430 Sherman Ave., Palo Alto, CA 94306. (415) 327-0125.

Fifteen computers, each doing its own thing, can be connected to a central disk file via the Cluster/One system.

Software

SAM 76/TRS-80. The SAM 76 language is now available for the TRS-80 in addition to other 8080/Z80 machines. The language is available on diskette or cassette and costs $15. For first class U.S. mail add $2; for overseas air book rate add $5. An update of the SAM 76 language manual is also available for $15 plus appropriate postage. Send orders to: SAM 76 Inc., Box 257, RR1, Pennington, NJ 08534. (609) 466-1130.

Audio BASIC. A BASIC programming course consisting of 12 cassette tapes coordinated with printed texts is now available from the Williamsville Publishing Company. This course allows students to hear in-depth explanations of the material in the text and to set their own pace. According to author William R. Parks, an assistant professor of information systems management at SUNY, Buffalo, the audio narrations were developed during actual computer runs. The BASIC tape 'n text course is divided into three modules: *Programming in BASIC, Intermediate BASIC,* and *Advanced BASIC,* each consisting of four cassette tapes with four printed texts. In the first part it is assumed that the student has no previous programming experience. The entire course (three modules) sells for $58. Single modules are $19.95 each. Order from: Williamsville Publishing Company, Box 237, Williamsville, NY 14211.

Computer-Dial. A new program from Michigan's Software Exchange enables you to turn your TRS-80 into a telephone dialer, using an interface circuit built with $4-worth of parts from Radio Shack. The Z80 Telephone Dialer Program is designed for the TRS-80 Level I, 4K micro. Twenty phone numbers may be dialed using the letters 'A' through 'T' for access. The program is of particular use to the handicapped and the elderly. The Z80 Telephone Dialer Program is available on cassette tape for $7.95 and includes complete instructions, with interface circuit diagram. Write: Software Exchange, 2681 Peterboro, W. Bloomfield, MI 48033.

HIRES Graphics. This collection of programs for the Apple II facilitates that computer's high resolution graphics capabilities. The set includes: Software-Controlled Character Display (allowing you to display lower case, APL, Russian, Japanese, mathematical notations or any characters one chooses); Character Set Generator and Editor (allows user to define and edit character sets, even invent a new language); Shape Vector Table Assembler and Editor (enables user to edit, create, store and display HIRES forms); Find Utility (returns current position on screen); Look Utility (looks at point on screen and determines if a point is plotted there). The minimum implementation requirements are 4K Apple II integer BASIC and standard HIRES graphics routines (INIT, SHAPE, POSN, CLEAR). HIRES Graphics Utility Set costs $9.95 from the Soft One, 315 Dominion Drive, Newport News, VA 23602.

MICROMONOPOLY. You can play Monopoly on North Star disk systems with this new program written in North Star Release 4 BASIC. Two versions are available: one for Solos/Cutter rating monitor and a non-Cutter version. Please specify machine type when ordering. Program sells for $25. Micro Business Systems, P.O. Box 15995, Tampa, FL 33684. (813) 885-4107.

Bridge Challenger. With this program you can play four-person Contract Bridge against the computer. The program, designed for 8K PETs, 16K Level II TRS-80s, and 16K Apples, will deal hands at random or according to your criterion for high card points. You can save hands on cassette and reload them for later play, as well as review tricks, rotate hands East-West, shuffle only the defense hands, or replay hands when the cards are known. Bridge Challenger is available for $14.95 from Personal Software, P.O. Box 136, Cambridge, MA 02138.

More Checkers. Officially called 8080 Checkers, this program will run on any 8080/Z80 computer with 12K RAM and a memory-mapping display, e.g., the TRS-80, SOL, VDM-1. 8080 Checkers can be set to play at two levels of difficulty (four or six moves ahead). At level four, the program responds in less than four seconds; at level six, it's usually less than 60. The software is available on CUTS cassette tape and North Star diskette. Prices are $19.50 and $24.50 from TCD Incorporated, P.O. Box 58742, Houston, TX 77058.

Multi-User BASIC. There's now a multi-user capability for Cromemco computers. Up to seven users can run BASIC programs independently with the new Cromemco software system, which provides an efficient, low-cost alternative to traditional time-sharing. This new capability has been accomplished through a memory-bank-select feature on Cromemco memory boards, high-performance disk drives, and software composed of a time-sharing operating system and BASIC. Multi-User BASIC is supplied as software only or as a combination hardware/software package to upgrade a Cromemco System Two or System Three to a two-user system. For additional information, contact Cromemco, Inc., 280 Bernardo Avenue, Mountain View, CA 94043. (415) 964-7400.

Conferences

West Coast Computer Faire. May 11-13. San Francisco Civic Auditorium. For more information, see *RC*, March-April 1979, or write: Computer Faire, Box 1579 Palo Alto, CA 94302. (415) 851-7075.

DINO-CON. June 16-17. Dunfey's Royal Coach Inn, San Mateo, CA. Movies, contests, D & D, big dealers row, computer modulated games and special guests from the worlds of SF-science and television. For information: DINO-CON, 22195 Redwood Road, Castro Valley, CA 94546. (415) 538-3918.

SIGPC '79. The first annual conference on Research and Development in Personal Computing will be held August 8-10, 1979, in Chicago at the Hyatt Regency O'Hare. The conference is sponsored by the Association for Computing Machinery (ACM) and its Special Interest Group on Personal Computing (SIGPC).

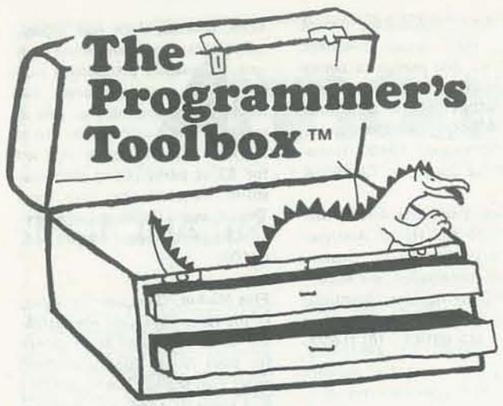
SIGPC '79 will be held during Chicago Computer Visualization Week (August 6-10, 1979) along with the IEEE Pattern Recognition and Image Processing Conference (PRIP-79) and the ACM/SIGGRAPH Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '79). A large trade show of personal computer and graphics equipment is planned to accompany papers, panels, user group meetings, workshops, and person-to-person poster booths. For more information, contact Maxine D. Brown, SIGGRAPH '79 Exposition, Hewlett-Packard, 19400 Homestead Road, Cupertino, CA 95014.

Northeast Computer Show. September 28-30. Hynes Auditorium, Prudential Center, Boston. For more information, see March-April *RC* or contact: Northeast Expositions, Box 678, Brookline Village, MA 02147. (617) 522-4467.

Other

Yankee Resources. A directory of microcomputer products and services in New England is now available from the Boston Computer Society. *The First New England Microcomputer Resource Handbook* contains information on computers, peripherals, software, retailers, repair organizations, courses, clubs, user groups, user publications, and trade journals. It enables prospective purchasers to compare costs and features of computers as well as services and support by local vendors. The handbook will sell for $2 at participating computer stores. Available by mail from The Boston Computer Society, 17 Chestnut Street, Boston, MA 02108.

Flea Market. Computer hobbyists in the New York City area gather the third Sunday of every month for their highly successful Computer Flea Market. Computer Flea is a show of computer programs and computer equipment for home and personal use. In addition to exhibits by vendors, inventors, and other entrepreneurs, each Flea features several talks aimed at computer novices. For more information, contact show director Robert Schwartz at 375 Riverside Drive, New York, NY 10025. (212) 663-5549 (evenings) or (212) 770-1333 (days).



BY EVERYBODY

In Vol. 1, No. 3 of PC, *1973, Marc LeBrun began a column that provided users with small, compact routines that could become part of the user's "toolbox" of computer skills.*

With this issue, we are reviving that column. We solicit material for this column from all of you.

In the future, we will begin to rewrite each subroutine, standardizing variable names and trying to make the routines reasonably independent of your main program modules. —RZ

PT1: WARP DRIVE

How about a routine that gives your TRS-80 Star Trek program a view-through-the-port during warp operations? It can be used in other programs as well by calling it with a GOSUB. Just be sure not to lose data from the main program.

```
800 '*** GRAPHIC "WARP DRIVE" DISPLAY
810 '*** BY MILAN D. CHEPKO    119 BELLEVILLE CRT
820 '*** THIEF RIVER FALLS, MN  56701
830 '*** CAN BE ADDED TO ANY "STARTREK" PROGRAM AS THE
840 '*** OBJECT OF A 'GOSUB'.  WILL DISPLAY UP TO 10
850 '*** STARS AT A TIME, ALTHOUGH 3 OR 5 SEEM BEST
990 DEFINT A-Z : DIM A(20)
1000 CLS
1010 FOR I=1 TO 20 : A(I)=RND(128) : NEXT I
1015 '*** OUTLINE VIEWING PORT
1020 FOR X=0 TO 127 : SET(X,0) : SET(X,47) : NEXT X
1030 FOR Y=0 TO 47 : SET(0,Y) : SET(127,Y) : NEXT Y
1035 '*** DETERMINE TIME SPENT IN THIS LOOP ***
1040 FOR N=0 TO 200
1045 '*** DETERMINE NUMBER OF STARS DISPLAYED (LIMIT=10)
1050 FOR I=1 TO 5
1055 '*** RECALL STAR'S LAST POSITION
1060 X=A(I):Y=A(I+10)
1065 '*** WILL NEXT POSITION BE BEYOND VIEWING PORT?
1070 IF X>1 AND X<126 AND Y>0 AND Y<47 GOTO1110
1075 '*** ERASE OLD STARS FROM SIDES OF VIEWING PORT
1080 IF X=1 OR X=126 THEN  RESET(X,Y)
1085 '*** FIND A NEW STAR NEAR CENTER OF SCREEN
1090 X=RND(90):IF X<38 GOTO1090
1100 Y=RND(36):IF Y<12 GOTO1100 ELSE GOTO1140
1105 '*** MOVE STAR TO NEXT POSITION ON SCREEN
1110 RESET(X,Y)
1120 IF X<64 THEN X=X-2 ELSE X=X+2
1130 IF Y<24 THEN Y=Y-1 ELSE Y=Y+1
1140 SET(X,Y)
1145 '*** STORE NEW LOCATION OF STAR
1150 A(I)=X:A(I+10)=Y
1160 NEXT I  : NEXT N
1165 GOTO1040
1170 CLS:END '***RETURN TO ORIGINAL PROGRAM
```

BY MILAN D. CHEPKO

PT2: APPLE SCAN SIMULATION

Here is a short Applesoft II program that simulates a high-resolution PPI scan. This capability may be useful in various games. However, the addition of appropriate blips and bleeps to the display is left as an exercise for the reader, since this will depend on the specific application.

```
100 REM *** PPI SCAN SIMULATION ***
110 REM
120 PI = 3.14159 : R = 95 : HCOLOR = 3
130 HGR2 : REM DRAW A CIRCLE
140 FOR A = 0 TO 2*PI STEP .02
150 HPLOT 140+R*SIN(A), 95+R*COS(A)
160 NEXT A : R = 94 : REM START SCAN
170 FOR B = 0 TO -2*PI STEP -.05
180 FOR I = 3 TO 0 STEP -3
190 HCOLOR = I
200 HPLOT 140,95 TO 140+R*SIN(B), 95+R*COS(B)
210 NEXT I,B
220 GOTO 170
```

BY JIM DAY