

\$2.50

In Canada \$3.00

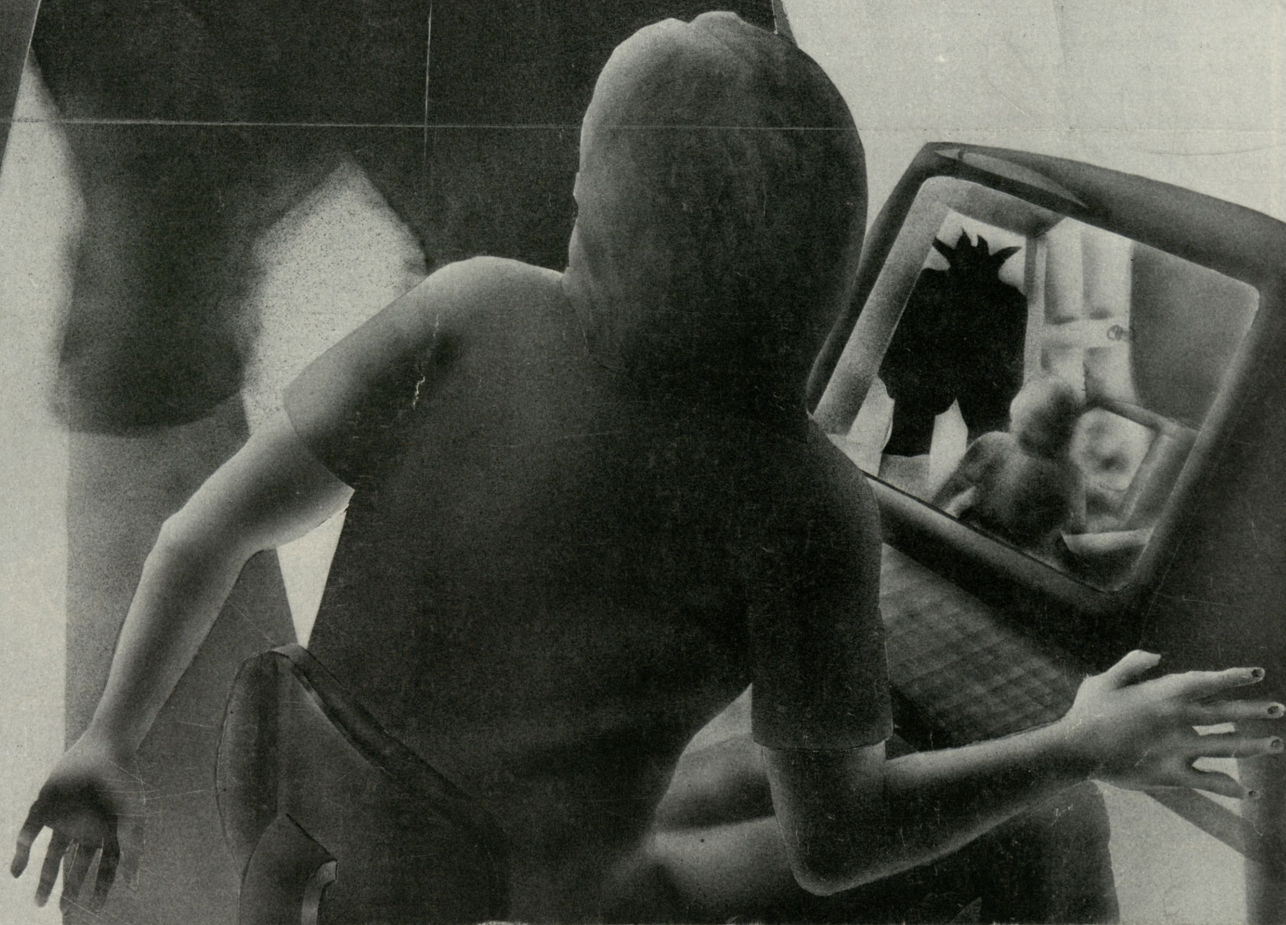
# Recreational COMPUTING

FOR THE IMAGINATIVE SMALL COMPUTER USER!

VOL. 9 NO.1 ISSUE 46

JULY - AUGUST 1980

FANTASY  
GAMES ISSUE!





# AIR TRAFFIC

In **Air Traffic Controller** you assume responsibility for the safe flow of

air traffic within a 15x25 mile area up to 5,000 feet in altitude. During your shift as a controller in charge of this airspace, 26 aircraft become active and under your control. Jets and prop planes have to be guided to and from the two airports, navigational beacons and ten entry/exit fixes. The aircraft enter the controller's airspace at various altitudes and headings whether or not you are ready.

**Air Traffic Controller** retains the basic realism of air traffic control. This program requires the same steady nerves under pressure and the same instant, almost instinctive, analyses of complex emergencies which are demanded of a professional air traffic controller. But "ATC" adds the excitement and well-defined goals of a game. This is just a simulation, and all passengers left in air traffic limbo by a panicked player will live to fly another day.

Your goal is to get all of the aircraft to their assigned destination before the shift is completed. At your disposal are a radar display of the aircraft positions in the control area; coded information concerning aircraft heading, destination and fuel supply; nav aids enabling you to hold aircraft or assign them automatic approaches; and commands to alter the altitude and heading of the aircraft. Working against you are altitude and heading requirements, fuel restrictions and, of course, the inimitable clock.

# CONTROLLER

The most obvious measure of difficulty of a game is the clock setting at the beginning. In a 99 minute game you will have time to go fix a sandwich between the appearance of two successive aircraft, while in the 16 minute game you may not have time to swallow before all of the aircraft have appeared.

No two games, even at the same clock setting, are ever alike. As controller, you must cope with the unique requirements of each aircraft. The game will end if you commit a "boundary error," that is, if an aircraft fails to leave your area at the proper altitude and exit fix...causing an unpleasant surprise for the controller next door. The game also ends if you

fail to leave a comfortable margin of safety between the aircraft as they whiz past each other. In cases of excessive delay, fuel supply considerations will become invested with a particular sense of urgency.

Successful guidance of all aircraft to their destination is a heady accomplishment. This never fails to thrill ATC enthusiasts at each successive level of play.

Your local retail store should carry Creative Computing Software. If your favorite retailer does not carry the software you need, have him call in your order to (800) 631-8112. Or you can order directly from Creative Computing. Write to Creative Computing Software, P.O. Box 789-M, Morristown, NJ 07960. **Air Traffic Controller** is now available for the 16K TRS-80 (3006), for the 16K Apple II and Apple II Plus (4008), the 8K Sorcerer (5008) and for the 4K Sol-20 (8001). All are on cassette for \$9.95. Include \$1.00 for postage and handling. For faster service, call in your bank card order toll free on our order hotline, (800) 631-8112.

Prices are subject to change without notice.

**creative  
computing  
software**

**sensational software**



## Editors' Notes

Our focus this issue is Computer Fantasy and Simulation Games. We have so much material that we've had to skimp on space for some of our regular features. CFSG's have been a recurrent theme in *Recreational Computing* beginning with an article Lee Hoevel and I wrote for the March-April 1978 issue titled "Epic Computer Games." Since that time we've seen these games migrate from large machines to small and become widely available from commercial suppliers. John Morrison provides an overview of the CFSG's today and provides some history, background, and pointers to the literature. "Wizard's Castle" by Joseph E. Power and "The Wonderful World of Eamon" by Donald R. Brown are complete CFSG's. Both are worthy of study and play. Furman Smith has provided us with a tutorial on making your own CFSG. His program is really designed to be modified and extended into another game of your own construction. Serg Korem tackles the problem of making a dialogue based Star Trek game in a tutorial. Lastly, I have a short note speculating on the impact future technology might have on CFSG's.

NCC—the computer industry held its giant gala trade show in Anaheim towards the end of May. It was a major success if one counts heads; there were over 80,000 people there. On occasion one could not negotiate the aisles forward, backwards, sideways, or whatever.

It used to be that the personal computing people had a separate identity from the "real" computer people. This year that distinction seems to have disappeared; one found the personal computer people and the old guard mini and maxi people side by side in the exhibit hall.

We're still looking for new authors, new ideas. Your letters and comments are always welcome.

We made a couple of printing gaffs in the last issue. Some of the material printed with a color graphic over was difficult if not impossible to read. Many readers were kind enough to point it out; one even mentioned that he thought that *Interface Age* was the only magazine to do that sort of thing and was sad to see us trying to compete with them. We will be more careful in the future. Jim Day's article, "A Proposed Graphics Language" is reprinted in this issue in a readable form. Our apologies to those of you who were outraged!

## ADVERTISING

ADVERTISING space is available in this publication. Please direct inquiries to the Advertising Director, People's Computer Company, Box E, Menlo Park, CA 94025. (415) 323-3111.

## SUBSCRIPTION INFORMATION

### U.S. RATES

- ☐ \$12/1 yr. (6 issues)
- ☐ Retaining subscription @ \$25 (\$13 tax deductible)
- ☐ Sustaining subscription @ \$100+ (\$98 + tax deductible)
- ☐ Lifetime subscription @ \$900+ (\$700 tax deductible)

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

### BACK ISSUES

\$2.50 each:  
Vol 7, Nos. 1, 2; Vol. 6, Nos. 1, 2, 3, 4, 5  
Outside the U.S. add \$.50 per issue.

### FOREIGN RATES

Payments must be in \$US drawn on a US bank.  
☐ \$20 Canada First Class  
☐ \$26 Rest of World Airmail  
☐ \$16 World Surface Mail

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

\$3.00 each:  
Vol. 7, Nos. 3, 4, 5, 6  
Vol. 8, No. 1, 2, 3, 4, 5, 6

## Staff

**Editor**  
Dennis Allison  
**Contributing Editors**  
Bob Albrecht  
Tracy Deliman  
Don Inman  
Ramon Zamora  
**Copy Editor**  
Sara Le Forge  
**Publisher**  
Ann Merchberger  
**Art Director/Production Manager**  
Carole Cullenbine  
**Artists**  
Fred Carpenter  
Carolyn Miller  
Carol Sevilla  
**Typesetters**  
Phyllis Adams  
Renny Wiggins  
**Proofreader**  
Nancy Heubach  
**Marketing Director**  
Gavin W. Cullen  
**Advertising Director**  
Michael Madaj  
**Circulation Manager**  
Peter Clark  
**Circulation Assistant**  
Leah Dansby  
**Accounting Supervisor**  
Giuliana Peter

## Retaining Subscribers

Algorithmics, Inc.  
DeWitt S. Brown  
Gerald Bowman  
Robert Conners  
David R. Dick  
Mark Elgin  
Joi Ellis  
John B. Fried  
Scott B. Guthery  
Alan Hamilton  
Brian Herring  
T. Alton Howard  
William G. Hutchison, Jr.  
W. A. Kelley  
William M. Richman II  
Phillip A. Smith  
Neil Sullivan  
Joseph A. Weisbecker  
Brett Wilson

## Sustaining Subscribers

Bill Godbout Electronics  
Byte Publications  
Paul, Lori and Tom Calhoun  
Louis R. Patzke



Cover by Fred Carpenter

# Recreational COMPUTING

July/August 1980  
Volume 9, Issue 1

## FANTASY GAMES ISSUE

- 4 WRITE YOUR OWN COMPUTER FANTASY SIMULATION  
*by Dr. Furman H. Smith*
- 10 WIZARD'S CASTLE  
*by Joseph R. Power*
- 18 YOU, TOO, CAN FIGHT DRAGONS  
*by John M. Morrison*
- 23 STAR TREK, A DIALOGUE APPROACH  
*by Serg Korem*
- 27 ON FUTURE FANTASY GAMES  
*by Dennis Allison*
- 30 CRYPTARITHMS  
*by John Davenport Crehore*
- 32 THE WONDERFUL WORLD OF EAMON  
*by Donald Brown*
- 46 CRYPTARITHMS FROM WAY BACK  
*by John Davenport Crehore*
- 48 TRS-80 ART?  
*by Ken Hahn*
- 50 MACHINES OF LOVING GRACE — A CELEBRATION  
*by David Thornburg*
- 51 IN DEFENSE OF HACKERS  
*by Steven Kudlak*
- 52 TOUCH, PANELS AND INTERACTIVE GRAPHICS  
*by Jim Day*
- 54 ANNOUNCEMENTS
- 56 A PROPOSED GRAPHICS LANGUAGE  
*by Jim Day*
- 58 PROGRAMMING PROBLEMS & SOLUTIONS  
*by Bob Albrecht, Don Albers & Jim Conlan*
- 62 FORTRAN MAN  
*by L. Schneider & T. Voros*

*Recreational Computing* (ISSN #0164-5846) 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, non-profit corporation. Donations are tax-deductible. Second class postage paid at Menlo Park, California, and additional entry points. Copyright 1980 by People's Computer Company, Menlo Park, California.

**FOREIGN DISTRIBUTORS OF RECREATIONAL COMPUTING:** UK & EUROPE: L P Enterprises, 8/11 Cambridge House, Cambridge Road, Barking, Essex, IG11 8NT, Great Britain. Hofacker-Verlag, Tegernseer Strasse 18, D-8150 Holzkirchen, West Germany. Computerland/Computer Store AB, Box 7134, Kungsgatan 19, S-10387 Stockholm, Sweden. CANADA: RS-232, 186 Queen Street W., Toronto, Ontario M5V 1Z1, Canada. ASIA & AUSTRALIA: Electronic Concepts Pty Ltd., 55 Clarence Street, Sydney, NSW 2000, Australia. Computer Store, POB 31-261, 22 B Milford Road, Milford, Auckland 9, New Zealand. ASCII Publishing, 305 Hi Torio, 5-6-7 Minami Aoyama, Minato-ku, Tokyo 107, Japan. Systems Formulate Corporation, Shin-makicho Building, 1-8-17 Yaesu, Chuo-ku, Tokyo 103, Japan. AFRICA: Computers, Etc., 170 Hendrik Verwoerd Drive, POB 51978, Randburg 2125, Republic of South Africa.



BY DR. FURMAN H. SMITH

## Invitation

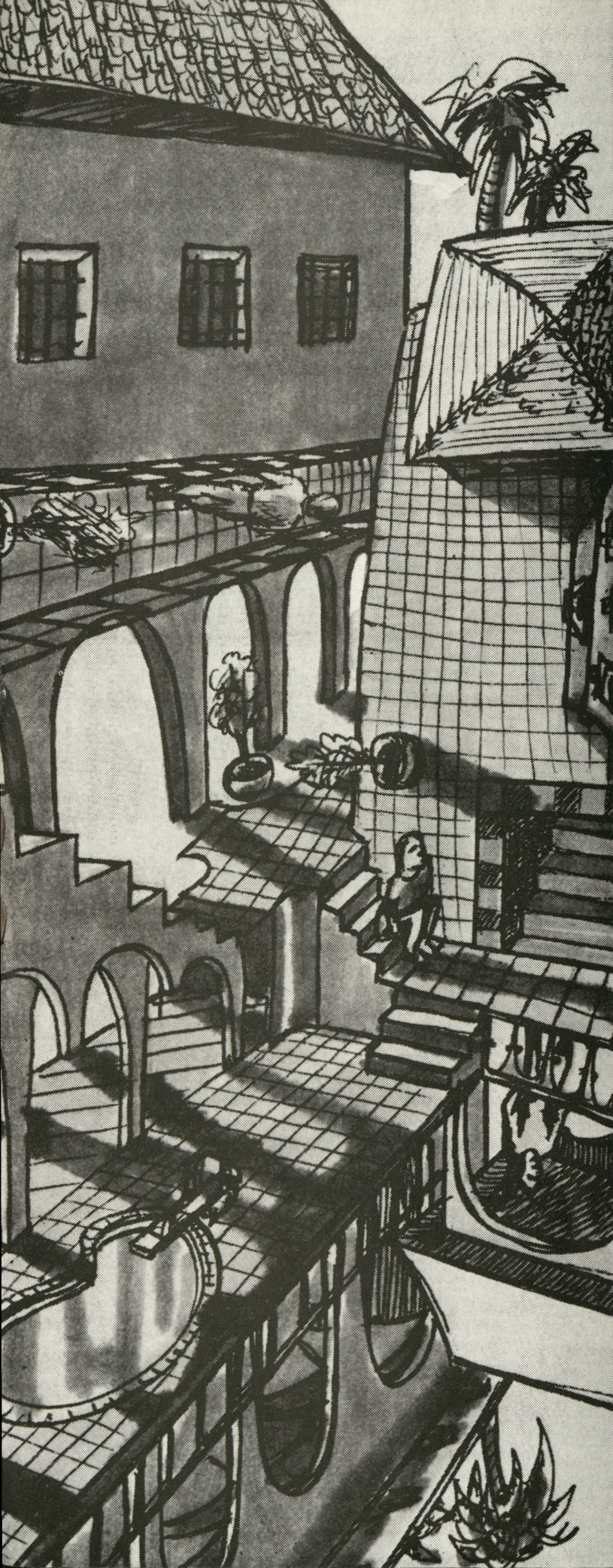
The *Nellan is Thirsty* version of *The Enchanted House* computerized fantasy simulation (CFS) is intended for children to play. The program *Enchanted House* is general and was designed for easy modification. The present article describes the program and gives ways of making simple modifications; a later article will go into major changes.

## Flavor of this CFS

As with any CFS, obstacles and unfavorable situations must be overcome to win. The *Nellan is Thirsty* version is nonviolent and the creatures are all friendly; thus, planning, problem solving, and inventiveness are encouraged in a nonthreatening atmosphere.

The player indicates choice of actions by messages such as GO NORTH, UNLOCK THE DOOR, TAKE THE EGG. Only the first and last words of the messages are considered; short forms such as SOUTH and S are accepted. There are HELP, RELOOK, and LIST WORD commands. Possible commands are often suggested by all-cap words in the descriptions; for example, "There is a magic MAP here. If you GET the MAP, you may later CONSULT the MAP." In this article, I will follow the convention of capitalizing every letter of words that the program understands and messages from humans.

The magic MAP draws itself: explored rooms are labeled, their walls and doorways appear on the MAP (untried doorways as asterisks, previously tried doorways as open or blocked, according to their state on the last attempted passage); unexplored areas are shown by question marks. (While drawing a map associated with playing a CFS is a good learning experience, it is beyond the capabilities of many children who would otherwise be able to enjoy and benefit from the game; a finished map would rob them of some of the joy of exploration.)





# your own computer fantasy simulation

The pathways are rectangular and usually intuitive; that is, movement is N, S, E, W, U, and D and if going NORTH from the Bank Room takes us to the White Room, then going SOUTH from the White Room will take us to the Bank (provided the passage is open). Some passages may be one-way; there may be poles that we can slide DOWN but not climb UP. Of course, magic also provides a means of transportation in the Enchanted House.

The adventurer's name is used often. If Everett is in the Bank, a giant carpet on the floor says "WELCOME EVERETT." CHULA, the white RABBIT, will greet Armenta with, "Well, my friend ARMENTA, I hope that you are enjoying your adventure."

The team of adventurers is to earn points by DEPOSITing (or DROPPing) treasures in the Bank Room. If there is more than one adventurer, each gets a turn of, for example, four scoots (scoot = one passage from one room to another). A clock in the Bank announces the time (in scoots) and a player may CONSULT her or his WATCH at any time. The game ends when all the treasures have been DEPOSITed or a certain number of scoots have elapsed.

## Major Program Ideas and Notation

Things like the MAP, CAT, and COIN would usually be termed objects, but I chose to call them manifestations since some objects appear in several forms; for example, manifestation number 13 is a CAT giving the adventurer a VASE and manifestation 14 is the same CAT napping. The list of manifestations is given on lines 6200-6299 of the program.

The location of manifestation M is denoted by L(M). The information that the COUPON (M=2) is in room 5 (the Hot Room) is summarized by

L(2)=5. The value minus one is used when the adventurer is carrying the object; if the adventurer is carrying the COUPON, then L(2)=-1. The value zero is used when a manifestation doesn't currently exist. For example, if the BOWL with cold MILK (M=4) is taken in the Hot Room, the location of manifestation 4 is changed from -1 to 0 since the MILK would no longer be cold, and the location of the BOWL of warm MILK (M=5) is changed from 0 (not existing) to -1 (being carried); this transformation is handled by line 2063 of the program. The 0 and -1 conventions for L(M) are from Scott Adams' 1979 article in *Creative Computing*.

When the adventurer gives his message (example: GRAB THE COIN), the message is split into a verb part (V\$="GRAB") and a noun part (N\$="COIN"). For reasons of space and efficiency, numbers V and N, rather than character strings V\$ and N\$, are used to process the message. Reference to the list of verbs, nouns, and manifestations (lines 6200-6299) shows that "GRAB THE COIN" yields V=3 and N=18.

Now let's look at the listing of variables (lines 6000-6199). The number of verbs that the program understands is denoted by V9. Leaving V9 as a parameter (as opposed to a constant such as 20) makes the program easier to modify. Similarly, N9 is the number of nouns the program understands and M9 is the number of manifestations found in the game.

The "9" following a variable means (in the system of notation I've no doubt pirated from some source I've forgotten) the upper limit on that variable; thus if C denoted the number of objects that the adventurer is carrying, C9 denoted the maximum number of objects that can be carried. The current time is denoted by T and the game is over if T=T9. Don't look at line 6078 yet - what's the notation for the number of rooms in the house?

I will leave the W-vectors for the next article, but if anyone wants to build his/her own Enchanted House right away, he/she should study the block 1600-1799 in conjunction with lines 6034-6056.

Some nouns describe more than one manifestation; thus, there cannot be a one-to-one correspondence between nouns and manifestations. Line 6122 tells us that N1(M)=N means that the noun number N describes the manifestation number M, for examples, N1(2)=13, N1(3)=12, N1(4)=12.

Sometimes we have more than one noun for a single manifestation; BANKER (N=23) and KLINKOYN (N=130) both describe the ninth manifestation. This difficulty is overcome in general by transforming to the lowest appropriate value and in particular by changing an N of 30 to an N of 23. The same trick is used for verbs; for example, a V of 5, 6, 7, or 8 will be transformed into a V of 5.

## Minor Remodeling of the Enchanted House

The program *Enchanted House* was designed so that it would be easy to build other Enchanted Houses. However, there is the worry expressed in the superb Lebling, Blank, and Anderson article, "Zork: A Computerized Fantasy Simulation Game" (July-August 1979 issue of *Recreational Computing*), although in a different context: "There is no such thing as a small change." I hope that the *Enchanted House* program allows minor changes, and I invite you to make several changes before we study major changes in a forthcoming issue.

Perhaps you'd like to rename a character. It only takes a change in line 1230; for example, to name the CAT "KAIFA," replace "NELLAN" in line 1230 by "KAIFA."

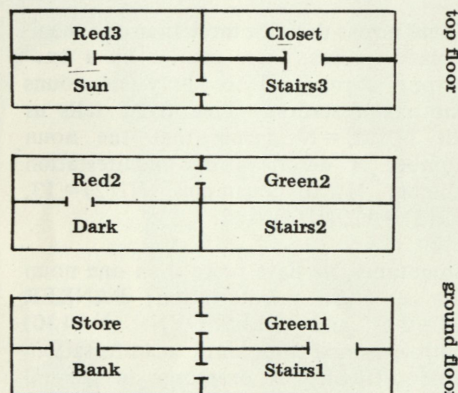


Are you a stamp collector? Change the word "VASE" to "STAMP" in line 1225 and change the description in the 2649-2658 block. If the STAMP can be said to be small, that's all the modification you need make; otherwise, the only other thing that you need to do is to delete the word "small" in line 2536.

### Homework for the Intrepid

Another CFS, *Deposit the Chair*, is described below. A forthcoming issue of *Recreational Computing* will have an article on modifying the current program to implement *Deposit the Chair*. The point of presenting the CFS an issue early is that some of you might want to take the modification (of the program in this issue) as a problem to work before seeing a solution in the next issue.

The *Deposit the Chair* version uses the following three-storey two-by-two map:



A CHAIR worth 30 points is initially in the Dark Room. The Red3 Room has the top of a red fireperson's pole and Red2 has the bottom of the pole. Similarly, the green rooms house a green pole. The adventurer can slide DOWN a pole under any conditions but can climb

UP if and only if she or he is not carrying the CHAIR. Similarly, the CHAIR doesn't fit through the small doorway between Green1 and Stairs1. The DOOR between the Store and Green1 is locked. One needs a SUPERBROOM to get past the COBWEB between Red1 and Green2.

The SUPERBROOM is in the Closet and the KEY is obtained by DROPPing the COUPON on the Store. Green2 is a nice initial location for the COUPON; put the COIN and JEWEL anyplace except the Bank.

I used the following values for V, N, and M. The first four manifestations are (1) machine with KEY, (2) machine without (4) Green2 Room side of COBWEB; the remaining manifestations (of M9 = 11) are identified by the noun list below and the fact that for  $M \geq 5$ ,  $N1(M) = M + 14$ . The  $V9 = 17$  verbs, in order, are TAKE, GET, GRAB, PICK, DROP, GIVE, OFFER, DEPOSIT, OPEN, UNLOCK, LIST, WORTH, HELP, GO, RELOOK, CONSULT, SWEEP. The  $N9 =$

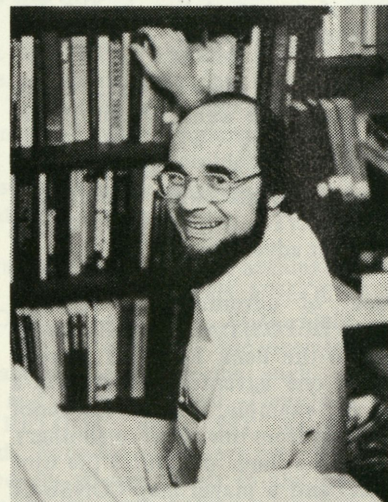
25 nouns, in order, are: N, S, E, W, NORTH, SOUTH, EAST, WEST, U, D, UP, DOWN, INVENTORY, SCORE, WORDS, WATCH, DOOR, COBWEB, COUPON, KEY, SUPERBROOM, MAP, COIN, JEWEL, CHAIR.

If you attempt this exercise, be sparing in your descriptions. Text takes a lot of memory.

### Documentation and Dissemination

Documentation is very expensive in terms of storing the program. My 48k system will not hold both BASIC and the program printed in this article. Fortunately, I have a single disk system (a Heath H89 "All-in-One") and there's plenty of room on a disk. I wrote a program I call *Hack and Pack* which takes a program as input, hacks out all comments, deletes unnecessary blanks, and thus produces an output program that is difficult to read but which can be stored in much less space. An example line of output is found in line 35 of the program printed here.

*Furman Smith received a Ph.D. in Probability and Statistics from the Florida State University in 1972, taught three years at the University of Kentucky, and has since been at the University of Houston Victoria Campus. He is currently an Associate Professor of Mathematical Sciences teaching four courses, Chairperson of the UHUC Faculty Council, and a member of numerous committees including an eight faculty member group that is advisory to the President of the University of Houston System. He has a marvelous wife, two marvelous kids, one good home computer, a garden, and backlog of work.*



## Enchanted House

### Nellan is Thirsty

12 / Furman H Smith  
14 / University of Houston Victoria Campus  
15 / 2302-C Red River  
16 / Victoria, Texas 77901

20 / This program appears in the  
21 / July-August 1980 issue of RECREATIONAL COMPUTING.

25 / Copies for the Radio Shack level II 16 computer-k TRS-80  
26 / will be available from RECREATIONAL COMPUTING.

28 / Copy submitted to Heath User's Group for HDOS MBASIC  
29 / (c) users with 16 computer-k bytes above MBASIC.

31 / Warning: The program as written here will not fit in  
32 / my 48 computer-k system. The program that I run has  
33 / all comments and unnecessary blanks deleted; for ex.,  
34 / this line isn't in the program and line 135 looks like  
35 / 135 IFP<1 THEN PRINT "Oh, come on.":GOTO130  
36 / Radio Shack users should omit the THEN in such lines.

50 / 0- 99 giving credits and contents  
51 / 100- 199 initializing  
52 / 200- 299 describing locations & manifestations  
53 / 300- 399 determining player's actions  
54 / 400- 499 ordinary movins, settins, & droppins  
55 / 500- 599 otherwise respondins  
56 / 600- 699 updatsins

```

57 / 700- 799 instructing Player
58 / 800- 999 (blank)
59 / 1000-1999 details of initializins. In particular:
60 / 10xx introducins values of parameters & DIM
61 / 11xx introducins verbs
62 / 12xx introducins nouns
63 / 13xx intro. graphics for rectangular floor plan
64 / 14xx intro. graphics for possible doorways
65 / 15xx introducins rooms
66 / 16xx describins N-S doorways
67 / 17xx describins E-W doorways
68 / 18xx into. certain properties of manifestations
69 / 19xx initializins other than above
70 /
71 / 2000-2399 details of describins rooms
72 / 2400-2999 details of describins manifestations
73 / 3000-3999 respondins to unlisted words
74 / 4000-4499 details of ordinary movins, settins, & droppins
75 / 4500-4999 drawins the map
76 / 5000-5999 details of otherwise respondins
77 / 6000-6199 listins variables found in the program <-- note
78 / 6200-6399 listins verbs, nouns, & manifestations <-- note
79 /
80 / ----- initializins -----
81 /
82 / What does line 105 do? See lines 106-109.
83 / CLEAR 250 :GOSUB 705 :GOSUB 1010 :GOSUB 745
84 / CLEAR 250 allocates extra string space
85 / GOSUB 705 puts some instructions on the screen
86 / GOSUB 1010 rushes off & initializes
87 / GOSUB 745 hops back & interacts with the human
88 / INPUT "Are you the only person playins (YES or NO) "; A$
89 / P = 1 : A$ = LEFT$(A$,1) : IF A$="V" THEN P = 1 : GOTO 140

```



```

120 IF A$<>"u" AND A$<>"n" THEN 130
125 PRINT "Please use capitol letters." : GOTO 110
130 INPUT "How many of you will be playing this time " : P9
135 IF P9<1 THEN PRINT "Oh, come on." : GOTO 130
140 IF P9>8 THEN A$ = "initials " ELSE A$ = "name "
145 DIM P$(P9) : I=1 : PRINT "Please tell me your " : A$ : INPUT P$(P)
150 IF I = P9 THEN 165 ELSE I = I + 1
155 PRINT "Please give me the " : A$ : " of the next player."
160 INPUT P$(I) : PRINT "Thank you." : GOTO 150
164
165 IF P9=1 THEN T2=T9 ELSE T2=4
169
200 ----- describing locations & manifestations ----
201
202 / Radio Shack level II TRS-80 users should change line
203 / 205 to read CLS : H89 & H19 users should remove the first
204 / comment apostrophe.
205 / PRINT ESC$;"E" / clears the Heath H89, H19 screen
207
208 / the current room R is described starting on line 2000+10*R
209 W4(R) = 1 : I = 1 + INT((R-1)/R8) : J = R - R8*(I-1)
210 ON I GOSUB 220, 225, 230 : GOTO 235
215 ON J GOTO 2010, 2020, 2030
220 ON J GOTO 2040, 2050, 2060
225 ON J GOTO 2070, 2080, 2090
234
235 FOR M = 1 TO M9
240 IF L(M) = R THEN GOSUB 265
245 NEXT M
248
249 / TRS-80 users should replace RND(1) by RND(0) everywhere.
250 IF RND(1) > T*(2-W4(1)-W4(3))/(<?T*9) THEN 260
255 PRINT "A voice says, <<Type HELP if you need help>>."
260 GOTO 305
262
263 / Description of manifestation M starts on line 2400+10*M
265 PRINT : I = 1 + INT((M-1)/8) : J = M - 8*(I-1)
270 ON I GOTO 275, 280, 285, 290
275 ON J GOTO 2410, 2420, 2430, 2440, 2450, 2460, 2470, 2480
280 ON J GOTO 2490, 2500, 2510, 2520, 2530, 2540, 2550, 2560
285 ON J GOTO 2570, 2580, 2590, 2600, 2610, 2620, 2630, 2640
290 ON J GOTO 2650
299
300 ----- determining player's actions ----
301
305 PRINT : PRINT "***** " : Q$(1 + Q9 * RND(1)) :
307 / The I=FRE(A$) in 310 to reclaim some scratch string space--
308 / at least in the Radio Shack version of Microsoft (c).
310 I=FRE(A$) : IF RND(1) > .5 THEN PRINT " " : P$(P) :
315 PRINT "?" : PRINT TAB(8) : INPUT A$ : I9 = LEN(A$) : PRINT
320 IF I9 = 0 THEN 305
324
325 FOR I=1 TO I9 : IF MID$(A$,I,1) <> " " THEN NEXT I
330 U$ = LEFT$(A$,I-1) : IF I = I9 + 1 THEN N$ = U$ : GOTO 345
335 FOR J = 19 TO I STEP -1 : IF MID$(A$,J,1) <> " " THEN NEXT J
340 N$ = RIGHT$(A$, I9-J)
344
345 U = 0 : N = 0
350 FOR I=1 TO U9 : IF U$(I) = U$ THEN U=I : GOTO 355 ELSE NEXT I
355 FOR I=1 TO N9 : IF N$(I) = N$ THEN N=I : GOTO 357 ELSE NEXT I
356
357 IF <N AND N<9 THEN 405 / allows SCOOT QUICKLY NORTH , etc
360 IF U=N OR <U AND U<9 AND U=N THEN 380 ELSE 3505
364
365 PRINT "I don't understand " : A$ : "
370 PRINT "Type LIST WORDS to see a list of"
375 PRINT "some of the words that I understand." : GOTO 305
379
380 IF N<29 THEN 405 ELSE IF N=29 THEN N=11 ELSE N=N-7
399
400 ----- ordinary movins, settings, & droppins ----
401
405 IF N>4 AND N<9 THEN N = N-4 / N=1 N=2 N=3 N=4 ~
409 / N E U&D
410 IF N>4 OR N=0 THEN 435 ELSE ON N GOTO 412, 416, 420, 424 / ~
411
412 W1(R) = ABS(W1(R)) : IF W1(R) = 1 THEN 430
414 IF W1(R) = 2 THEN R = R - R8 : GOTO 605 ELSE 505
415
416 W1(R+R8) = ABS(W1(R+R8)) : IF W1(R+R8) = 1 THEN 430
418 IF W1(R+R8) = 2 THEN R = R + R8 : GOTO 605 ELSE 505
419
420 W2(R+1) = ABS(W2(R+1)) : IF W2(R+1) = 1 THEN 430
422 IF W2(R+1) = 2 THEN R = R + 1 : GOTO 605 ELSE 505
423
424 W2(R) = ABS(W2(R)) : IF W2(R) = 1 THEN 430
426 IF W2(R) = 2 THEN R = R - 1 : GOTO 605 ELSE 505
427
430 PRINT "You can't walk through walls." : GOTO 305
434
435 IF N<12 AND N<22 THEN GOSUB 4015 ELSE GOSUB 4030
439
440 IF ( U<4 OR U=13 ) AND N=22 THEN 5410
444
445 IF M<0 OR U=0 OR U>8 THEN 455
450 PRINT "That's not possible now." : P$(P) : " : GOTO 305
454
455 IF M=0 OR U=15 OR N=9 THEN 505
460 IF <U AND U<9 THEN 475
464
465 IF L(M)>-1 AND L(M)<>R THEN 470 ELSE 475
470 PRINT "It seems that the " : N$ : " isn't here." : GOTO 305
474
475 IF U>8 THEN 505 ELSE IF U<5 THEN U=1 ELSE U=5
480 IF N=28 THEN PRINT "The WATCH stays on your wrist." : GOTO 305
485 IF N=10 OR N=23 OR N=24 THEN 5980 / UNTAKEable manifestations
489
490 ON (U+3)/4 GOSUB 4045, 4100
495 GOTO 235
499
500 ----- otherwise responding ----
501
505 IF N <= 8 AND N>0 THEN 5015
509
510 IF N=26 OR N=27 THEN ON (N-25) GOSUB 5105, 5155 : GOTO 305
512 IF U=0 THEN 365
514
515 IF U <= 14 THEN J = U-8 : GOTO 520 ELSE J = U-14 : GOTO 525
520 ON J GOTO 5205, 5255, 5355, 5355, 5405, 5405, 5605
525 ON J GOTO 5705, 5755, 5780, 5805, 5980, 210
529
530 ----- updating ----
531
535 T=T+1 : T1=T1+1
540 IF T>T9 THEN GOSUB 5155 ELSE 625
544
545 PRINT : PRINT "Time is up -- the " : T9 : " scoots have passed."
550 PRINT : INPUT "If you want to play again, push return." : A$ : RUN
554
555 IF W1(R)=-1 THEN W1(R)=1
556 IF W1(R+R8)=-1 THEN W1(R+R8)=1
557 IF W2(R)=-1 THEN W2(R)=1
558 IF W2(R+1)=-1 THEN W2(R+1)=1
564
565 IF T1 <= T2 THEN 205 ELSE T1=0 : IF P<P9 THEN P=P+1 ELSE P=1
568 PRINT " # " : "It's " : P$(P) : "'s turn now." : PRINT : GOTO 210
569
570 ----- instructing Player ----
571
575 PRINT CHR$(27) ; "E" / to clear screen (see line 205)
579
580 PRINT "Let's Pretend that you are in an enchanted house."
585 PRINT
586 PRINT "You should use one or two words to tell me what you"
587 PRINT "want to do. I'll suggest messages at first."
588 PRINT : PRINT "If you have a question while you're inside " :
589 PRINT "the house." : PRINT "Type HELP." : RETURN
594
595 PRINT : PRINT "When you have read all this, Please"
596 PRINT "push the button that says RETURN (or ENTER)." : TAB(59) :
597 INPUT A$ : PRINT : RETURN
599
600 ----- details of initializing ----
601
605 /--- introducing values of parameters and DIMensioning
609
610 DEFINT A-U / Just ignore if your BASIC doesn't have
614
615 R6=1 : R7=3 : R8=3 : R9=R6+R7+R8 : R0=R9-R8+1 : R=R0
616 U9 = 20 : N9 = 31 : M9 = 25
617
618 DIM U$(U9), N$(N9), L(M9), N1(M9), W(M9), W4(R9), W4$(R9)
619 DIM W1((R6+R7+1)+R8), W2(R9+1), W1$(3), W2$(3), W3$(9)
624
625 T=0 : T1=0 : T9=72 : C=0 : C9=4
627
628 / Radio Shack level II TRS-80 users should change line
629 / 1040 to read RANDOM.
630 I=RND((PEEK(8219))) / to seed the HDOS MBASIC random # gen
634
635 Q9=4 : B9=4 : DIM Q$(Q9), B$(B9)
639
640 /line 1055 is for Heath graphics on the H17 & H89
641 / (VG$ for Yes Graphics & NG$ for No Graphics)
642 ESC$ = CHR$(27) : VG$ = ESC$ + "F" : NG$ = ESC$ + "G"
643
644 /--- introducing verbs
645
646 FOR I = 1 TO U9 : READ U$(I) : NEXT I
647 DATA TAKE, GET, GRAB, PICK, DROP, GIVE, OFFER, DEPOSIT
648 DATA KICK, CONSULT, OPEN, UNLOCK, FILL, LIST, WORTH, SIT
649 DATA DRINK, HELP, GO, RELOOK
654
655 /--- introducing nouns
656
657 FOR I = 1 TO N9 : READ N$(I) : NEXT I
658 DATA N, S, E, W, NORTH, SOUTH, EAST, WEST, DOOR, CHAIR
659 DATA RABBIT, BOWL, COUPON, WRAPPER, MAP, KEY, EGG, COIN
660 DATA JEWEL, CATALOG, VASE, MILK, BANKER, CAT, WORDS
661 DATA INVENTORY, SCORE, WATCH, CHULA, KLINKOVN, NELLAN
664
665 /--- introducing graphics for the rectangular floor plan
666
667 / Line 1320 would work (when not as a mere comment) on any
668 / CRT with +s. I use lines 1335,1340 as noncomments
669 / to work on my Heath H89. If you have a level II TRS-80
670 / with more than 16 k, you should omit or make into
671 / comments lines 1335-1340 and remove the comment
672 / apostrophies from lines 1345-1360. For other computers,
673 / keep line 1320 until you work out the graphics & memory.
674
675 FOR J=1 TO 9 : W3$(J) = "+++" : NEXT J
676
677 / Lines 1335-1340 are for the Heath H17 & H89.
678
679 FOR J=1 TO 9 : READ A$ : W3$(J)=VG$+A$+NG$ : NEXT J
680 DATA " ea","aua","ad "," va","aba","at "," fa","asa","ac "
681
682 / Lines 1345-1360 are for the Radio Shack level II TRS-80
683 / with over 16 computer-k.
684
685 FOR J=1 TO 9 : READ I1,I2
686 W3$(J) = CHR$(I1) + CHR$(I2) : NEXT J
687 DATA 138,141, 142,141, 142,133, 170,157, 174,157
688 DATA 174,149, 168,156, 172,156, 172,148
689
690 /--- introducing graphics for possible doorways
691
692 / If you would like to use symbols that would work on most
693 / CRT's, remove the comment apostrophes from lines
694 / 1420, 1430, 1440, 1450, 1460 and omit (or make into
695 / comments) the other lines in this 1400-1499 block.
696
697 FOR I=0 TO 3 : READ W1$(I), W2$(I) : NEXT I
698
699 DATA "++ **,+," " * "
700
701 / untried doorways
702
703 DATA "+++++++", "+++"
704
705 / walls (not doorways at all)
706
707 DATA "++ ++", " "
708
709 / doorways that were clear last time tried
710
711

```



```

1460 DATA "++#-#++", "#-#"
1461 / doorways that were blocked last time tried
1469 /
1470 / Lines 1470-1475 are for the Heath H19 & H89.
1471 /
1472 / W1$(0) = VG$+"aaa"+NG$: W2$(0) = VG$+" * "+NG$
1473 / W1$(1) = VG$+"aaaaaa"+NG$: W2$(1) = VG$+" * "+NG$
1474 / W1$(2) = VG$+"aaa"+NG$: W2$(2) = VG$+" * "+NG$
1475 / W1$(3) = VG$+"aaaaa"+NG$: W2$(3) = VG$+" * "+NG$
1476 /
1477 / Lines 1482-1487 are for Radio Shack level II TRS-80
1478 / computers with over 16 computer-k of RAM.
1481 /
1482 / A$ = CHR$(140) + CHR$(140) + CHR$(140)
1483 / W1$(0) = A$+"* "+A$: W2$(0) = " * "
1484 / W1$(1) = A$+LEFT$(A$,2)+A$: W2$(1) = CHR$(170)+CHR$(149)
1485 / W1$(2) = A$+" "+A$: W2$(2) = " "
1486 / W1$(3) = A$+CHR$(152)+CHR$(164)+A$
1487 / W2$(3) = CHR$(162) + CHR$(132)
1499 /
1505 / --- introducing rooms
1506 /
1510 FOR I=0 TO 9 : READ W4$(I) : W4(I)=0 : NEXT I
1514 /
1515 DATA " ? "
1520 DATA " Office ", " Milk ", " Store "
1525 DATA " White ", " Gold ", " Hot "
1530 DATA " Bank ", " Green ", " Cat "
1599 /
1605 / --- describes N-S possible doorways
1606 / (clear doorways, locked doors, walls, blocked doors)
1607 /
1610 FOR I = 1 TO (R6+R7+1)*R8 : READ W1(I) : NEXT I
1613 /
1614 / line 1615 describes the North wall
1615 DATA 1, 1, 1
1616 /
1617 / Office Milk Store
1618 / line 1620 describes 3 things:
1619 / Office door, clear doorway between Milk & Gold, and wall
1620 DATA -2.5, -2, -1
1621 / White Gold Hot
1622 / line 1625 describes 3 things:
1623 / clear doorway between White & Bank, wall between Gold &
1624 / Green, and the clear doorway between Hot & Cat
1625 DATA -2, -1, -2
1626 / Bank Green Cat
1627 / line 1630 describes the South wall
1630 DATA 1, 1, 1
1699 /
1705 / --- describing E-W doorways
1706 /
1710 FOR I = 1 TO R9+1 : READ W2(I) : NEXT I
1714 /
1715 DATA 1, -1, -2.5
1720 DATA 1, -2, -2
1725 DATA 1, 1, -2, 1
1799 /
1805 / --- introducing certain properties of manifestations
1806 /
1810 FOR M=1 TO M9 : READ L(M), N1(M) : W(M) = 0 : NEXT M
1815 DATA 1.20, 6.13, 0.12, 0.12, 0.12, 2.9, 3.0, 0.0, 7.23
1820 DATA 1.16, 9.24, 0.24, 0.24, 0.24, 0.11, 0.11, 0.11, 4.11
1825 DATA 7.15, 2.14, 1.0, 0.18, 0.17, 3.19, 0.21
1830 FOR I = 22 TO 25 : W(I) = 25 : NEXT I
1899 /
1905 / --- initializing other than above
1906 /
1910 FOR I = 1 TO 09 : READ Q$(I) : NEXT I
1915 DATA "What are you going to do now", "What now"
1920 DATA "What do you choose to do", "What is your decision"
1924 /
1925 FOR I=1 TO B9 : READ B$(I) : NEXT I
1930 DATA happy, "glad to see you", busy, sleepy
1995 RETURN
1999 /
2000 / ----- details of describing rooms -----
2003 /
2009 / R=1 => Office
2010 PRINT "Well, "; P$(P); ", you are in the Office."
2011 PRINT "You are standing next to a CHAIR which is next to a"
2012 PRINT "desk in a room with yellow walls and brown carpet."
2013 PRINT "This is a room of study and joy."
2014 PRINT "There is a door in the south wall." : RETURN
2019 / R=2 => Milk Room
2020 PRINT "This is the Milk Room. If you have a BOWL, you can"
2022 PRINT "say FILL BOWL or GET MILK and have your BOWL"
2023 PRINT "filled with nice cold MILK. There are doorways to"
2024 IF L(15)=-1 AND L(6)=2 THEN L(18)=0 : L(17)=2
2025 PRINT "the south and east." : RETURN
2029 / R=3 => Store
2030 PRINT "You are in the Store." : RETURN
2038 /
2039 / R=4 => White Room
2040 PRINT "Now you are in the White Room--the walls are white."
2041 PRINT "There is a wall to the west. You know that the Bank"
2042 PRINT "is through the south doorway. There's a door to"
2043 PRINT "the north and a doorway to the east." : RETURN
2049 / R=5 => Gold Room
2050 PRINT "All the walls in this room are painted gold."
2052 PRINT "There are exits to the north, east, and west."
2053 IF L(15)=-1 AND L(23)=0 THEN 2054 ELSE RETURN
2054 L(15)=0 : L(16)=5 : L(23)=5 : RETURN
2059 / R=6 => Hot Room
2060 PRINT "Oh it's hot in here. You are in the Hot Room and if"
2061 PRINT "cold MILK were in this room, it would be warmed."
2062 PRINT "The MAP shows the doorways."
2063 IF L(4)=-1 THEN L(4)=0 : L(5)=-1 : RETURN ELSE RETURN
2068 /
2069 / R=7 => Bank
2070 PRINT "You are in a room called the Bank. A giant carpet"
2071 PRINT "on the floor says WELCOME "; P$(P); "."
2075 PRINT "A screen says that you are invited to visit for IT9
2076 PRINT "scouts." : T1 scouts have passed. To leave this"
2078 PRINT "room, type NORTH or N or GO NORTH." : RETURN
2079 / R=8 => Green Room
2080 PRINT "You are standing in a room with green walls."
2082 PRINT "A green CHAIR is in the corner." : CHOP sign for space
2083 IF W1(4) < 0 THEN RETURN ELSE PRINT "A sign here says:"
2084 PRINT TAB(20); STRING$(25, 95)
2085 PRINT TAB(20); "!" : TAB(26); "Magic works"; TAB(44); "!"
2086 PRINT TAB(20); "!" : TAB(29); "in the"; TAB(44); "!"
2087 PRINT TAB(20); "!" : TAB(24); "Enchanted House." : TAB(44); "!"
2088 PRINT TAB(20); "!" : STRING$(23, 95); "!" : RETURN
2089 R=9 => Cat Room
2090 PRINT "You are in the room where "; N$(31); " lives; this"
2091 PRINT "room called the Cat room." : RETURN
2399 /
2400 / ----- details of describing manifestations -----
2408 /
2409 / M=1 => CATALOG
2410 PRINT "There is a college CATALOG from the University of"
2411 PRINT "Houston Victoria Campus here." : RETURN
2418 /
2419 / M=2 => COUPON
2420 PRINT "On the floor is a COUPON. According to what is"
2421 PRINT "written on the COUPON, if it is DROPPED in the Store"
2422 PRINT "while the machine is working, it will be good for one"
2423 PRINT "blue BOWL." : RETURN
2428 /
2429 / M=3 => empty BOWL
2430 PRINT "There is an clean empty blue BOWL here." : RETURN
2439 / M=4 => BOWL with cold MILK
2440 PRINT "The BOWL of cold MILK is here." : RETURN
2449 / M=5 => BOWL with warm MILK
2450 PRINT "There is a blue BOWL with warm MILK." : RETURN
2458 /
2459 / M=6 => carrot DOOR
2460 RETURN / Let the Carrot Door be a surprise.
2468 /
2469 / M=7 => machine with bowls
2470 PRINT "In a corner of the room is a machine which announces"
2472 PRINT "that if you DEPOSIT (or DROP) a COUPON and the"
2473 PRINT "machine is working, you will receive whatever the"
2474 PRINT "coupon is good for. If the machine doesn't work,"
2475 PRINT "you are invited to call the repairperson." : RETURN
2479 / M=8 => machine without bowls
2480 PRINT "The machine in the corner has a flashing display"
2481 PRINT "announcing <<out of BOWLS>>." : RETURN
2488 /
2489 / M=9 => BANKER
2490 PRINT "The BANKER, Mr. "; N$(30); ", looks ";
2492 PRINT B$(1+B9*RND(1)) ; ", "
2493 IF L(23)=R0 THEN PRINT "Type SCORE to check your score."
2494 RETURN
2498 /
2499 / M=10 => KEY
2500 PRINT "There is an old brass KEY here." : RETURN
2508 /
2509 / M=11 => CAT telling of wanting cold milk
2510 PRINT N$(31); " the CAT is here. She pretty-much fills up"
2511 PRINT "the room; she's as large as an elephant--no kiddins."
2512 PRINT "In a soft polite voice she says, << I'd love to have"
2513 PRINT "some nice cold MILK. Many of my friends have tried"
2514 PRINT "to deliver cold MILK here and many have failed."
2515 PRINT "Perhaps you, "; P$(P); ", would be kind enough"
2516 PRINT "to try. I must warn you: It will not be easy >>."
2517 RETURN
2519 / M=12 => CAT just after warm milk
2520 PRINT N$(31); " purrs and says, <<Thank you."
2522 PRINT P$(P); ", for OFFERING me this good MILK>>."
2523 PRINT "She doesn't mention that it was warm."
2525 L(5)=0 : L(3)=9 : L(12)=0 : L(11)=9 : RETURN
2529 / M=13 => CAT with cold milk
2530 PRINT "Wow! "; N$(31); " can sure be loud when she purrs."
2531 PRINT "She looks at you and says, << "; P$(P); ", you've"
2532 PRINT "made me so happy. I know that it wasn't easy to get"
2533 PRINT "this BOWL of cold MILK to me. You did it!"
2534 L(4)=0 : L(3)=9 : L(13)=0 : L(14)=9 : IF L(25)<0 THEN 2537
2535 PRINT "Please accept this small present >>. She offers you"
2536 PRINT "a small "; N$(21); ", " : L(25)=9
2537 RETURN
2539 / M=14 => napping CAT
2540 PRINT N$(31); " is napping. She's quite happy." : RETURN
2548 /
2549 / M=15 => usual RABBIT
2550 PRINT N$(29); " the white RABBIT is here and she says"
2551 PRINT "<<Well, my friend "; P$(P); ", I hope that"
2552 PRINT "you are enjoying your adventure>>." : RETURN
2559 / M=16 => RABBIT giving egg
2560 PRINT N$(29); " hops out of your arms and scurries off. She"
2561 PRINT "returns and says <<Please TAKE this golden egg that"
2562 PRINT "I hid here in the Gold Room>>."
2563 L(23)=5 : L(15)=5 : L(16)=0 : C=C-1 : RETURN
2569 / M=17 => RABBIT eating carrot door
2570 PRINT N$(29); " hops down and slobbers a growth of "
2571 PRINT "carrots that was blocking the east doorway. She "
2572 PRINT "smiles and says, << Thanks, friend "; P$(P); " >>."
2573 L(17)=0 : L(15)=2 : L(6)=0 : C=C-1 : RETURN
2579 / M=18 => initial RABBIT
2580 PRINT "A white RABBIT is here. She hops to you and says,"
2581 PRINT "<<I am "; N$(29); ", welcome to the Enchanted House,"
2582 PRINT "my friend. "; P$(P); ". She hugs you and says,"
2583 PRINT "<< I have a present saved for you in the room to"
2584 PRINT "the east of here. If you say TAKE RABBIT (or"
2585 PRINT "TAKE "; N$(29); " and next say GO EAST (or E or"
2586 PRINT "EAST), you can carry me to the Gold Room>>."
2587 L(18)=0 : L(15)=4 : RETURN
2588 /
2589 / M=19 => magic MAP
2590 PRINT "There is a magic MAP here. If you GET the MAP, you"
2591 PRINT "may later CONSULT the MAP." : RETURN
2598 /
2599 / M=20 => WRAPPER
2600 PRINT "An old gum WRAPPER is on the floor." : RETURN
2603 /
2609 / M=21 => reference and credit
2610 PRINT "A voice says, << Furman Smith's July 1980 ";
2611 PRINT "Recreational Computing"; PRINT "article helps ";
2612 PRINT "programmers remodel this Enchanted House. >>." : RETURN
2615 /
2619 / M=22 => COIN
2620 PRINT "A heavy COIN is here." : RETURN
2628 /

```

Continued on pg. 42



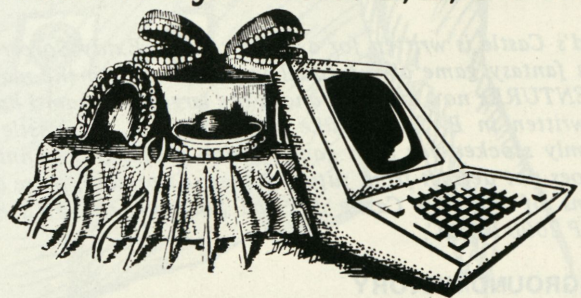
# USEFUL TALK FROM THE DOCTOR.

**New! DR. DOBB'S JOURNAL OF COMPUTER CALISTHENICS & ORTHODONTIA: Running Light Without Overbyte Vols. 1, 2, & 3**

(The People's Computer Company Series). Each volume includes almost everything from all of the issues of Dr. Dobb's Journal published in the years indicated.

**Vol. 1** contains the first ten issues and reflects the changes that took place in personal computing in 1976. Chapters include: A Realized Fantasy: Unlimited Speech Synthesis for Home Computers; A 6502 Disassembler from Apple: Timer Routines for 6800's; A BASIC Terminal Exchange Program; CHASE: A One or Two Player Video Game, and more.

**Vol. 2** (1977) chronicles the emergence of the small computer as a useful tool. It includes the source listing for Lawrence Livermore BASIC, the source code for Dr. Starkweather's 8080 Pilot, and much more. Chapters include: A KIM-1 TV Sketch Program; What to Get Your Computer for its First Birthday; More Support for Software Vendors; Some Dire Warnings of a Modern, and more.



**Vol. 3** (1978) looks at the growing interest in programming languages, along with articles on specialized applications and utilities. Chapters include: Memory Test for 6502; Relocation is Not Just Moving Programs; Z-80 RAM Tester; Absolutely Dazzling Dazzleware; Renumbering and Appending BASIC Programs on the Apple-II Computer; STRUBAL: A Structured BASIC Compiler; A KIM Binary Calculator; SAM 76 — Additional Comments, and more. **Vol. 1, #5475-0; Vol. 2, #5484-X; Vol. 3, #5490-4; \$18.95 each.**

**Available at your local computer store!**

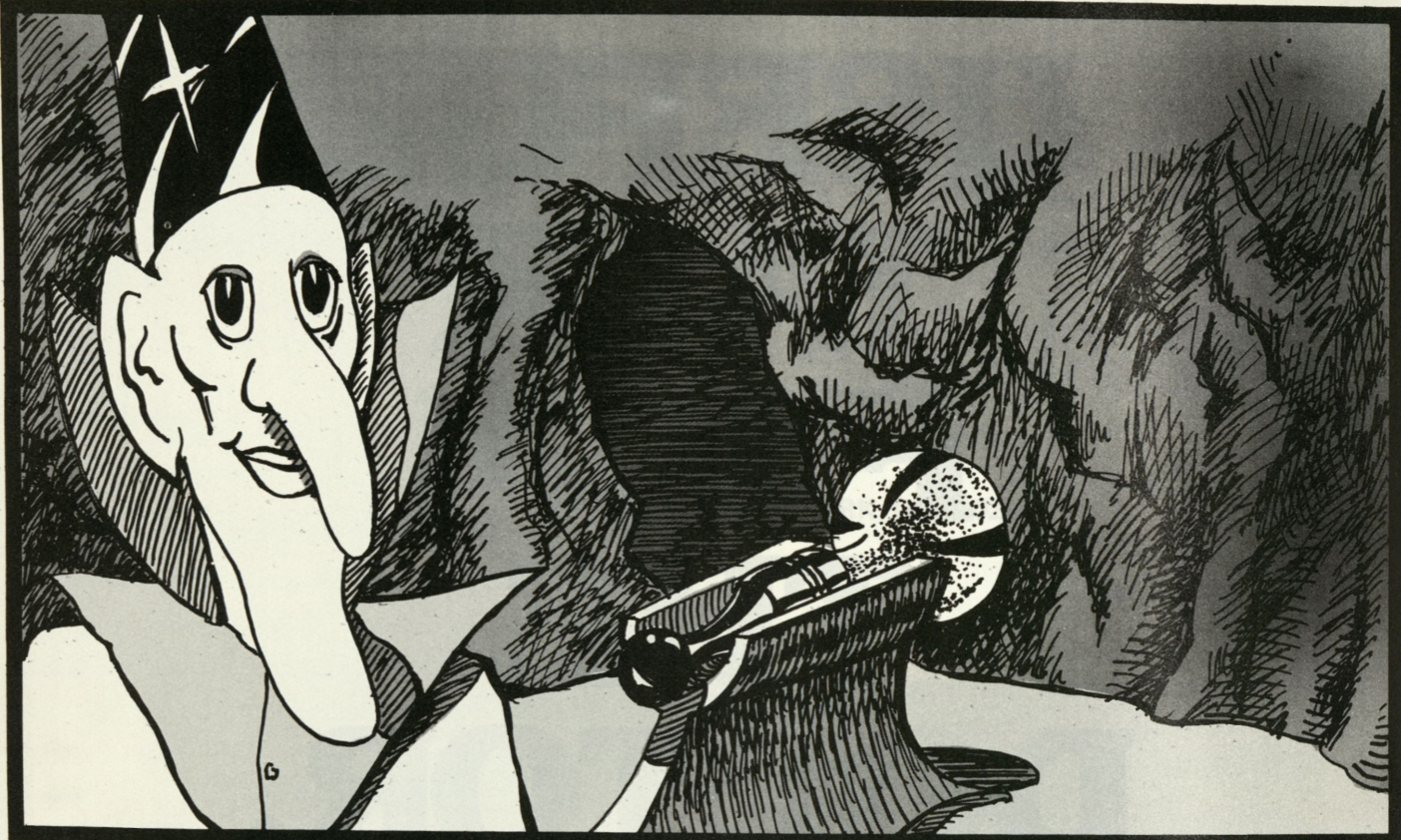


**HAYDEN BOOK COMPANY, INC.**  
50 Essex Street, Rochelle Park, N.J. 07662

**Call (201) 843-0550, ext. 307  
TO CHARGE YOUR ORDER TO  
Master Charge or BankAmericard!**

Minimum order is \$10.00;  
customer pays postage and handling.





# WIZARD'S CASTLE

BY JOSEPH R. POWER

*Wizard's Castle is written for a 16K or larger Exidy Sorcerer. It is a fantasy game about equal in complexity to the many ADVENTURES now available and is the largest program I have ever written in BASIC. Unlike ADVENTURE, the castle is randomly stocked for each game, which means that winning one does not detract from future games. At present, there are versions of Wizard's Castle for the Sorcerer, the TRS-80, and HP 2000 BASIC.*

## BACKGROUND HISTORY

Many cycles ago, in the kingdom of N'dic, the gnomish wizard Zot forged his great orb of power. Soon after this he vanished, leaving behind his vast subterranean castle filled with esurient monsters, fabulous treasures, and the incredible Orb of Zot. From that time hence many a bold youth has ventured into the wizard's castle. As yet, none has ever emerged victorious.

## INTRODUCTION

Wizard's Castle is a computerized simulation of one of the most common and popular fantasy motifs — the lone adventurer's quest within an immense underground labyrinth. Each game is separate from all others, so the game is a challenge even after you have won several times. Each game will result in a win or loss, depending on a player's skill and luck. The following instructions explain the rules and options of the game. If at any time, however, you are not sure of what to do — experiment. The program is designed to prevent invalid inputs.

## CHARACTER CREATION

At the start of each game you will be asked a number of questions about what type of character you will have. You must make choices about the following things:

**Race** — You may be an Elf, Dwarf, Man, or Hobbit. Each race starts with a total of 32 points (except Hobbits, who get 28) but they are distributed differently for each race.

**Sex** — You may be male or female. Both are equal in ability and number of points. Be creative with your response.

**Points** — Each character starts with a number of points for the attributes of Strength (ST), Intelligence (IQ), and Dexterity (DX). In addition, there are some other points which you may distribute to the three attributes as you wish.

Your ST, IQ, and DX may be any number from 1 to 18. If any of the three goes below 1, you have died. For all three attributes the bigger numbers are better. Each character also starts the game with 60 gold pieces (GPs) with which to purchase some, none, or all of the following:

**Armor** — You may buy Plate armor for 30 GPs, Chainmail for 20 GPs, or Leather for 10 GPs. You can wear only one suit of armor at a time. The more expensive the armor, the more damage it will absorb.

**Weapons** — You may buy a Sword for 30 GPs, a Mace for 20 GPs, or a Dagger for 10 GPs. You can carry only one weapon at a time. The more expensive the weapon, the more damage it will do to the various monsters.

**Lamp** — If, after selecting armor and a weapon, you have 20 GPs or more left, you may buy a lamp for 20 GPs. Having a lamp will allow you to look into adjacent rooms without



entering them.

**Flares** — If, after all other purchases, you have any money left, you may buy flares at 1 GP apiece. Lighting a flare reveals the contents of all the rooms around your current position.

Once you have created and equipped a character, you are ready to enter the castle and begin the game.

## THE CASTLE

The castle is arranged as an 8 x 8 x 8 matrix. This means that there are 8 levels with 64 rooms in each level. Each level of the castle is constructed like a donut in that the north edge is connected to the south edge and the east edge is connected to the west edge. In a similar fashion, the sinkholes (explained later) on level 8 will 'drop' you down to level 1. The ONLY room that does not work in this manner is the entrance/

O — a crystal orb (this is NOT the Orb of Zot)

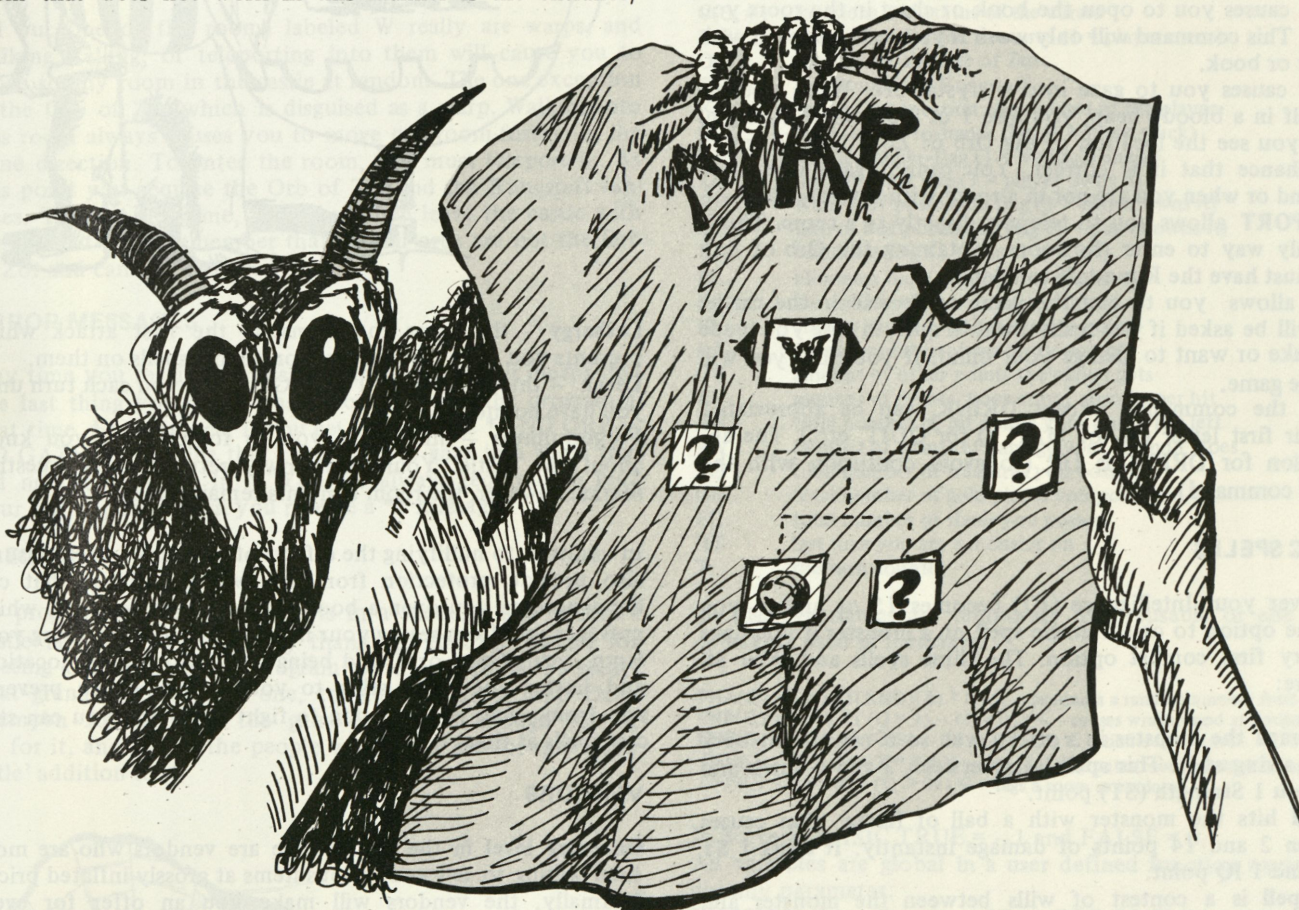
S — a sinkhole (a room with no floor)

T — one of the eight treasures.

The single letters are the abbreviations for the room contents which are displayed whenever you look at your map or light a flare. When you look at your maps, the room you are currently in is bracketed by <>.

## PLAYER COMMANDS

Whenever the program asks YOUR MOVE?, you must decide what action you wish to perform. If your choice is not valid in your current situation, the program will inform you and allow you to try again. The following is a list of the commands which the program understands, with a description of their effects and restrictions:



exit of the castle (always at (1, 4) Level 1). Going north from this room takes you out of the castle and ends the game.

Each room in the castle will have as contents one of the following:

- E — the entrance/exit of the castle, always located at (1,4) Level 1
- ◆ — an empty room
- U — stairs going up to the level above
- D — stairs going down to the level below
- P — a magic pool you can drink from
- C — a chest which may be opened
- B — a book which may be opened
- G — from 1 to 10 gold pieces
- F — from 1 to 3 flares
- W — a warp
- M — a monster (1 of 12 types)
- V — a vendor

**NORTH** moves you to the room north of your present position. **WHEN YOU GO NORTH FROM THE ENTRANCE THE GAME ENDS.** In all other cases the north edge wraps to the south.

**SOUTH** moves you to the room south of your present position. In all cases the south edge wraps to the north edge.

**EAST** moves you to the room east of your present position. In all cases the east edge wraps to the west edge.

**WEST** moves you to the room west of your present position. In all cases the west edge wraps to the east edge.

**UP** causes you to ascend stairs going up. You must be in a room with stairs going up.

**DOWN** causes you to descend stairs going down. You must be in a room with stairs going down.

**DRINK** causes you to take a drink from a magic pool. You may repeat this command as often as you wish, but you must be in a room with a magic pool.

**MAP** causes a map of the level you are currently on to be printed. All unexplored rooms are displayed as '?'; all other



rooms are displayed by their one-character abbreviations. You may look at your map at any time unless you are blind or retreating from a monster.

**FLARE** causes one of your flares to be lit, revealing the contents of all the rooms around your current position. Because each edge is joined to the opposite edge, you will always see nine rooms with your location in the center. Each flare you light burns away to nothing. You may light flares as long as you have some and if you are not blind or retreating from a monster. For people familiar with any of the common Star Trek games, a flare works much like a Long Range Scan with the addition of wraparound.

**LAMP** allows you to shine your lamp into any one of the rooms north, south, east, or west of your current position, revealing that room's contents. Unlike flares, a lamp may be used over and over again. You may use your lamp at any time unless you don't have a lamp, are blind, or are retreating from a monster.

**OPEN** causes you to open the book or chest in the room you are in. This command will only work if you are in a room with a chest or book.

**GAZE** causes you to gaze into a crystal orb. When you see yourself in a bloody heap, you lose 1 or 2 points of Strength. When you see the location of the Orb of Zot, there is only a 50% chance that it is correct. You cannot gaze when you are blind or when you are not in a room with crystal orb.

**TELEPORT** allows you to teleport directly to a room. This is the only way to enter the room containing the Orb of Zot. You must have the Runestaff to teleport.

**QUIT** allows you to end the game while still in the castle. You will be asked if you really want to quit, in case you made a mistake or want to change your mind. If you quit, you will lose the game.

All of the commands, except DRINK, can be abbreviated by their first letter (M for MAP, Q for QUIT, etc.). The abbreviation for DRINK is DR (to avoid confusion with the DOWN command).

## MAGIC SPELLS

Whenever your Intelligence (IQ) becomes 15 or higher, you gain the option to cast a magic spell on a monster if you have the very first combat option. The three spells and their effects are:

**Web** traps the monster in a sticky web so it can't fight back as you swing at it. This spell lasts between 2 and 9 turns and costs you 1 Strength (ST) point.

**Fireball** hits the monster with a ball of flame that causes between 2 and 14 points of damage instantly. It costs 1 ST point and 1 IQ point.

**Deathspell** is a contest of wills between the monster and yourself. Whoever has the lower IQ dies at once. It costs nothing to use but it is very risky. Even with an IQ of 18 (the highest possible), you have a 25% chance of losing.

## TREASURES, CURSES, BLINDNESS, AND SUCH

In the castle are eight randomly placed treasures:

**The Ruby Red** — wards off the curse of Lethargy.

**The Norn Stone** — has no special power.

**The Pale Pearl** — wards off the curse of the Leech.

**The Opal Eye** — cures blindness.

**The Green Gem** — wards off the curse of Forgetfulness.

**The Blue Flame** — dissolves books stuck to your hands.

**The Palantir** — has no special power.

**The Silmaril** — has no special power.

There are three curses:



**Lethargy** — this gives the monsters the first attack which prevents you from bribing them or casting spells on them.

**Leech** — this takes from 1 to 5 GPs from you each turn until you have no more.

**Forgetfulness** — this causes you to forget what you know about the castle. Your map slowly returns to all question marks; however, the room contents remain the same.

In addition to nullifying the effects of the curses, the treasures also provide protection from the two bad things that can happen when you open a book. These are going blind, which prevents you from seeing your maps, lighting flares, using your lamp, gazing into orbs, and being told your present location, and having the book stick to your hands, which prevents you from using your weapon to fight monsters (you can still cast spells at them though).

## VENDORS

On every level in the castle there are vendors who are more than willing to sell you various items at grossly inflated prices. Normally, the vendors will make you an offer for every treasure you have and then, depending on the amount of gold you have, will sell you new armor, a new weapon, ST, IQ, and DX potions (no matter how many of these you buy, 18 is the maximum value for all three), and a lamp if you don't already have one.

If you choose to attack a vendor, you will antagonize every vendor in the castle and they will react like a monster. You will also lose the ability to trade with them. Killing a vendor, however, will give you new plate armor, a sword, one ST, IQ, and DX potion, and a lamp (if you don't have one), in addition to his hoard of between 1 and 1000 GPs.

To end hostilities and reestablish trade, you must bribe any vendor in the castle with the treasure of his choice.

## MONSTERS AND THE RUNESTAFF

There are twelve types of monsters in the castle:



- |           |             |
|-----------|-------------|
| 1. Kobold | 7. Bear     |
| 2. Orc    | 8. Minotaur |
| 3. Wolf   | 9. Gargoyle |
| 4. Goblin | 10. Chimera |
| 5. Ogre   | 11. Balrog  |
| 6. Troll  | 12. Dragon  |

Please note that each time you hit a gargoyle or dragon there is a chance your weapon will break.

Each monster possesses a hoard of from 1 to 1000 GPs which you get when you kill it. In addition, one of the monsters is carrying the Runestaff (you won't know which one until you kill it). You must have the Runestaff to teleport, and when you teleport into the room containing the Orb of Zot, the Runestaff disappears.

## WARPS AND THE ORB OF ZOT

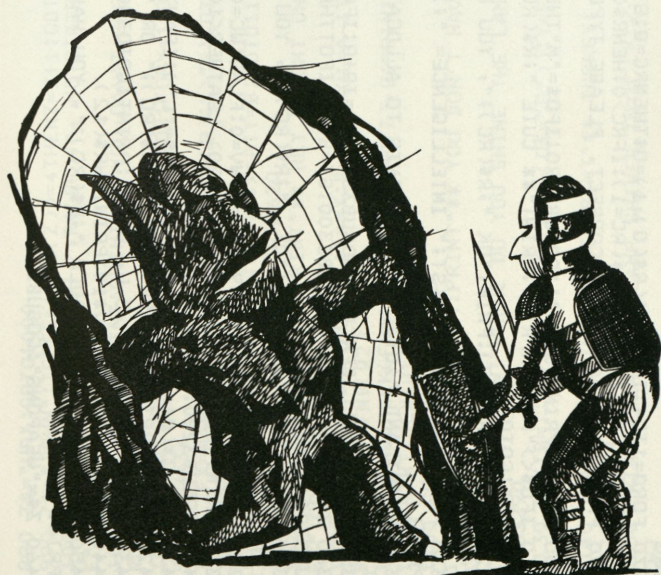
All but one of the rooms labeled W really are warps, and walking, falling, or teleporting into them will cause you to move to any room in the castle at random. The one exception is the Orb of Zot which is disguised as a warp. Walking into this room always causes you to move one room further in the same direction. To enter the room, you must teleport in. At this point you acquire the Orb of Zot and the Runestaff vanishes. To win the game, you must then leave the castle with the Orb of Zot. Remember that crystal orbs are not the Orb of Zot and cannot be picked up.

## ERROR MESSAGES

Any time you receive a message starting with \*\* it means that the last thing you typed was unacceptable to the program at that time. For instance, if you get the message \*\* NO ORB — NO GAZE, this means that you tried to gaze in a room that did not have an orb in it. You are always required to redo your last response when you receive a \*\* message.

## CREDITS

No program is created or exists in a vacuum, and Wizard's Castle is no exception. Many thanks go to Chip Bestler for creating the first Hobbit program which served as the great-great-grandfather to this one, to Kevin Williams and Dana Kaempfen for thinking the game was good enough to beg me for it, and to all the people who asked me for one more 'little' addition to it.



## FINAL INSTRUCTIONS

These instructions are meant to be a guide only. Feel free to experiment with various responses when running the game (this is half the fun of the game). For the best results, add equal parts of imagination and common sense. If you have any questions that you can't figure out, write to me and I will do my best to dispell confusion. Send all letters to Joseph R. Power, 124 Cedar St., Apt. 5, E. Lansing, MI 48823. Please include a self-addressed stamped envelope.

## Variables, Functions, and Machine Specifics for Wizard's Castle

CS(34)	names of all the possible castle contents
I\$(34)	abbreviations for all the possible castle contents
RS(4)	names of the four races
WS(8)	names of the four weapons and four armor types
ES(8)	names of the eight recipes (for Orc Tacos, etc.)
C(3, 4)	locations and status of the curses
T(8)	status flags for the treasures (1 = player owns)
O(3)	location of the Orb of Zot
R(3)	location of the Runestaff
X, Y, Z	usually the location coordinates of the player
BF	book-stuck-to-hands flag (1 = book stuck)
VF	Vendor-anger-flag (1 = Vendors angry)
LF	lamp-owned flag (1 = player owns it)
RF	Runestaff possession flag (1 = player owns it)
OF	Orb of Zot possession flag (1 = player owns it)
BL	blindness flag (1 = player is blind)
BL	blindness flag (1 = player is blind)
ST	current number of strength points
IQ	current number of intelligence points
DX	current number of dexterity points
OT	amount of other points the player gets
AV	number of points your armor absorbs per hit
AH	total number of hit points your armor has left
WV	number of points of damage your weapon does
TC	total number of treasures you possess
GP	total number of gold pieces you possess
FL	total number of flares you possess
HT	last turn you ate a monster on
T	the turn counter

all other variables are temporary and reusable or else string constants used in messages.

FNA(Q) = 1 + INT(RND)(8) \* Q <— produces a random number from 1 to Q  
 FNB(Q) = Q + 8 \* ((Q = 9) - (Q = 0)) <— causes wraparound at borders  
 FNC(Q) = -Q \* (Q < 19) - 18 \* (Q > 18) <— limits Q to a max. of 18  
 FND(Q) = Q \* 64 + X \* 8 + Y - 585 <— computes room location in memory  
 FNE(Q) = Q + 100 \* (Q > 99) <— tags a room as explored.

In Sorcerer BASIC TRUE = -1 and FALSE = 0.

All variables are global in a user defined function except the dummy parameter.

CHR\$(12) clears the screen like a TRS-80 CLS command.

The first remark is a machine language routine to simulate the RANDOM function.

After 32767, memory locations (for POKE and PEEK commands) are numbered -32768 (8000 hex) to -1 (FFFF hex).

# LISTING





# "WIZARD'S CASTLE"

```

10 REM _C2SLFF4
20 CLEAR60:DIMC$(34),I$(34),R$(4),M$(8),E$(8)
30 DIMC(3,4),T(8),O(3),R(3):PRINTCHR$(12);:C$="Cresting Arrays"
40 POKE260,218:POKE261,1:T=USR(0):T=PEEK(-2049)
50 DEFEND(Q)=Q*64+X*8+Y-585:DEFFNE(Q)=Q+100*(Q>99)
60 Y$=CHR$(13)+** ANSWER YES OR NO "+CHR$(13)
70 DEFFNA(Q)=1+INT(RND(8)*Q):DEFFNB(Q)=Q*8*(Q=9)-(Q=0)
80 Q=RND(-(.2*1+1)):RESTORE:FORQ=1T034:READC$(Q),I$(Q):NEXTQ
90 FORQ=-512T00:POKEQ,101:NEXTQ:FORQ=1T08:READW$(Q),E$(Q):NEXTQ
100 FORQ=1T04:READQ:(Q):NEXTQ:DEFFNC(Q)=Q*(Q>19)-18*(Q>18)
110 PRINTCHR$(12);:GOSUB3270:PRINTTAB(21);"THE WIZARD'S CASTLE":PR
INT:GOSUB3270
120 PRINT"COPYRIGHT (C) 1980 by Joseph R Power":PRINT
130 PRINT"Last Revised - 04/12/80 11:10 PM":PRINT
140 PRINT:PRINT"PLEASE BE PATIENT - ";X=1:Y=4
150 POKEFND(1),2:PRINT"IN";:FORZ=1T07:FORQ1=1T02:Q=104:GOSUB3200
160 POKEFND(241),103:NEXTQ1:NEXTZ:PRINT"IN";
170 FORZ=1T08:FORQ=113T0124:GOSUB3200:NEXTQ1:FORQ1=1T03
180 FORQ=105T0112:GOSUB3200:NEXTQ:Q=125:GOSUB3200:NEXTQ1:READO$;
PRINTO$;NEXTZ
190 FORQ=126T0133:Z=FNA(8):GOSUB3200:NEXTQ:PRINT"G";
200 Q=101:FORA=1T03:Z=FNA(8):GOSUB3200:C(A,1)=X:C(A,2)=Y:C(A,3)=
Z:C(A,4)=0
210 READO$:PRINTO$:NEXTA:PRINT"S";:RC=O:ST=2:DX=14:R$(3)="MAN"
220 Q=112+FNA(12):Z=FNA(8):GOSUB3200:R(1)=X:R(2)=Y:R(3)=Z
230 Q=109:Z=FNA(8):GOSUB3200:O(1)=X:O(2)=Y:O(3)=Z:PRINT"TL"
240 BF=O:OT=8:AV=O:HT=O:IT=1:VF=O:LF=O:TC=O:GF=60:RF=O:OF=O:BL=O
250 IQ=8:SX=O:FORQ=1T08:IT(Q)=O:NEXTQ:PRINTCHR$(12);:ALL RIGHT, B
OLD ONE"
260 PRINT:PRINT"YOU MAY BE AN ELF, DWARF, MAN, OR HOBBIT":GOSUB3
280
270 FORQ=1T04:IFLEFT$(R$(Q),1)=O$THENRC=Q:ST=ST+2*Q:DX=DX-2*Q
280 NEXTQ:PRINT:OT=OT+4*RC=1:IFRC>OTHER$(3)="HUMAN":GOTO300
290 PRINT** THAT WAS INCORRECT. PLEASE TYPE E, D, M, OR H.":GOT
0260
300 PRINT:PRINT"SEX ";:GOSUB3290:IFQ$="M"THENSX=1:GOTO320
310 IFQ$<>"F"THENPRINT:PRINT** CUTE ";R$(RC);", REAL CUTE. TRY
M OR F:GOTO300
320 PRINTCHR$(12);:PRINT"OK ";R$(RC);", YOU HAVE THESE STATISTICS
:";PRINT
330 PRINT"STRENGTH= ";ST;:INTELLIGENCE= ";IQ;:DEXTERITY= ";DX;
PRINT
340 PRINT"AND ";OT;:OTHER POINTS TO ALLOCATE AS YOU WISH.":PRIN
T
350 Z$="INTELLIGENCE ";GOSUB3300:IQ=IQ+Q:IFQ=O THEN370
360 Z$="STRENGTH ";GOSUB3300:ST=ST+Q:IFOT THENZ$="DEXTERITY ";GOS
UB3300:DX=DX+Q
370 PRINTCHR$(12);:PRINT"OK, ";R$(RC);", YOU HAVE 60 GOLD PIECES
(GP'S):PRINT
380 Z$="ARMOR":GOSUB3390:AV=O:WV=O:FL=O:WC=O
390 PRINT"PLATE<30> CHAINMAIL<20> LEATHER<10> NOTHING<0>"
400 GOSUB3280:IFQ$="N" THEN440
410 AV=-3*(Q$="P")-2*(Q$="C")-(Q$="L");:IFAV>O THEN440
420 PRINT:PRINT** ARE YOU A ";R$(RC);" OR ";C$(FNA(12)+12);
430 PRINT" ? TYPE P,C,L, OR N":PRINT:GOTO380
440 AH=AV*7:GP=GP-AV*10:PRINTCHR$(12)
450 PRINT:PRINT"OK, BOLD ";R$(RC);", YOU HAVE ";GP;" GP'S LEFT":
PRINT
460 Z$="WEAPONS":GOSUB3390

```

```

470 PRINT"SWORD<30> MACE<20> DAGGER<10> NOTHING<0>":GOSUB3280:IF
Q$="N" THEN500
480 WV=-3*(Q$="S")-2*(Q$="M")-(Q$="D");:IFWV>O THEN500
490 PRINT:PRINT** IS YOUR IQ REALLY ";IQ;"? TYPE S, M, D, OR N"
:PRINT:GOTO460
500 GP=GP-WV*10:PRINTCHR$(12);:IFGP<20 THEN540
510 PRINT"WANT TO BUY A LAMP FOR 20 GP'S ";:GOSUB3290
520 IFQ$="Y" THENLF=1:GP=GP-20:GOTO540
530 IFQ$<>"N" THENPRINT:PRINT:PRINT:GOTO510
540 PRINTCHR$(12);:IFGP<1 THENQ=0:GOTO600
550 PRINT:PRINT"OK, ";R$(RC);", YOU HAVE ";GP;" GOLD PIECES LEFT
":PRINT
560 INPUT"FLARES COST 1 GP EACH. HOW MANY DO YOU WANT ";O$
570 Q=VAL(O$):PRINT:IFQ>OORASC(O$)=48 THEN590
580 PRINT** IF YOU DON'T WANT ANY TYPE 0 (ZERO)":PRINT:GOT
0560
590 IFQ>G THENPRINT** YOU CAN ONLY AFFORD ";GP:PRINT:GOTO560
600 FL=FL+Q:GP=GP-Q:PRINTCHR$(12);:X=1:Y=4:Z=1
610 PRINT"OK ";R$(RC);", YOU ENTER THE CASTLE AND BEGIN.":PRINT:
GOTO1670
620 T=T+1:IFRF+OF>O THEN690
630 IF(C(1,4)>T(1)) THENT=T+1
640 IF(C(2,4)>T(3)) THENGP=GP-FNA(5):IFGP<O THENGP=0
650 IF(C(3,4)>T(5)) THEN690
660 A=X:B=Y:C=Z:X=FNA(8):Y=FNA(8):Z=FNA(8)
670 POKEFND(2),FNEE(PEEK(FND(2)))+100:X=A:Y=B:Z=C
680 IFPEEK(FND(3))=1 THENFORQ=1T03:C(Q,4)=-(C(Q,1)=X)*(C(Q,2)=Y)*
(C(Q,3)=Z):NEXTZ
690 IFNA(5)>1 THEN 790
700 PRINT:PRINT"YOU ";Q=FNA(7)+BL:IFQ>7 THENQ=4
710 ONGOSUB750,730,740,720,760,770,780:GOTO 790
720 PRINT"STEPPED ON A FROG":RETURN
730 PRINT"HEAR ";MID$( "A SCREAM FOOTSTEPSA WUMPUS THUNDER",FNA(4
)+9-8,9):RETURN
740 PRINT"SNEEZED":RETURN
750 PRINT"SEE A BAT FLY BY":RETURN
760 PRINT"SMELL ";C$(12+FNA(13));" FRYING":RETURN
770 PRINT"FEEL LIKE YOU'RE BEING WATCHED":RETURN
780 PRINT"ARE PLAYING WIZARD'S CASTLE":RETURN
790 IFBL+T(4)=2 THENPRINT:PRINTC$(29);:CURES YOUR BLINDNESS:BL=
0
800 IFBF+T(6)=2 THENPRINT:PRINTC$(31);:DISSOLVES THE BOOK":BF=0
810 PRINT:PRINT:PRINT:INPUT"YOUR MOVE ";O$;:IFLEFT$(O$,2)="DR"THE
N1180
820 O$=LEFT$(O$,1):IFO$="N" THEN 940
830 IF(O$="S" OR(O$="W" OR(O$="E" THEN 950
840 IFQ$="U" THEN 970
850 IFQ$="D" THEN 990
860 IFQ$="M" THEN1010
870 IFQ$="F" THENON1+BLGOTO1070,1010
880 IFQ$="L" THENON1+BLGOTO1110,1010
890 IFQ$="Q" THEN1310
900 IFQ$="G" THENON1+BLGOTO1480,1010
910 IFQ$="T" THENPRINT:ON1+RFGOTO1590,1600
920 IFQ$="Q" THEN1640
930 PRINT:PRINT** STUPID ";R$(RC);" THAT WASN'T A VALID COMMAND
":GOTO620
940 IFPEEK(FND(2))=2 THEN2900
950 X=X+(O$="N")-(O$="S");:Y=Y+(O$="W")-(O$="E")
960 X=FNB(X):Y=FNB(Y):GOTO1670
970 IFPEEK(FND(2))=3 THENZ=Z-1:GOTO1670

```







```

" ;
1860 GOSUB3290:IF0$="Y" THEN TC=TC-1:T(Q)=0:GP=GP+A:GOTO1880
1870 IF0$<"N" THEN PRINTY$:GOTO1850
1880 NEXT Q
1890 IFGP<1E3 THEN PRINT:PRINT"YOU'RE TOO POOR TO TRADE,";R$(RC);
GOTO1860
1900 IFGP<1250 THEN2130
1910 PRINT:PRINT"OK,";R$(RC);", YOU HAVE ";GP;" GOLD PIECES AND
";W$(AV+5)
1920 PRINT:Z$="ARMOR":GOSUB3390:PRINT"NOTHING<0> LEATHER<1250> "
;
1930 IFGP>1499 THEN PRINT"CHAINMAIL<1500> ";:IFGP>1999 THEN PRINT"PL
ATE<2000>";
1940 PRINT:GOSUB3280:PRINT:IF0$="N" THEN2010
1950 IF0$="L" THEN GP=GP-1250:AV=1:AH=7:GOTO2010
1960 IF0$="C" AND GP<1500 THEN PRINT"** YOU HAVEN'T GOT THAT MUCH CA
SH":GOTO1920
1970 IF0$="P" THEN GP=GP-1500:AV=2:AH=14:GOTO2010
1980 IF0$="M" AND GP<2000 THEN PRINT"** YOU CAN'T AFFORD PLATE":GOTO
1920
1990 IF0$="S" THEN GP=GP-2000:AV=3:AH=21:GOTO2010
2000 PRINT:PRINT"** DON'T BE SILLY. CHOOSE A SELECTION":GOTO1940
2010 IFGP<1250 THEN2130
2020 PRINT:PRINT"YOU HAVE ";GP;" GP'S LEFT WITH ";W$(WV+1);" IN
HAND":PRINT
2030 PRINT:Z$="WEAPON":GOSUB3390:PRINT"NOTHING<0> DAGGER<1250> "
;
2040 IFGP>1499 THEN PRINT"MADE<1500> ";:IFGP>1999 THEN PRINT"SWORD<2
000>";
2050 PRINT:GOSUB3280:PRINT:IF0$="N" THEN2130
2060 IF0$="D" THEN GP=GP-1250:WV=1:GOTO2130
2070 IF0$="M" AND GP<1500 THEN PRINT"** SORRY SIR, I DON'T GIVE CRED
IT":GOTO2030
2080 IF0$="S" THEN GP=GP-1500:WV=2:GOTO2130
2090 IF0$="S" AND GP<2000 THEN PRINT"** DUNGEON EXPRESS CARD - ";
2100 IF0$="S" AND GP<2000 THEN PRINT"YOU LEFT HOME WITHOUT IT!":GOTO
2030
2110 IF0$="S" THEN GP=GP-2000:WV=3:GOTO2130
2120 PRINT"** TRY CHOOSING A SELECTION":GOTO2050
2130 IFGP<1000 THEN620
2140 Z$="STRENGTH":GOSUB3370:IF0$<"Y" THEN2160
2150 GP=GP-1E3:ST=FNC(ST+FNA(6)):Q=ST:GOSUB3380:GOTO2130
2160 IF0$<"N" THEN PRINTY$:GOTO2140
2170 IFGP<1000 THEN620
2180 Z$="INTELLIGENCE":GOSUB3370:IF0$<"Y" THEN2200
2190 GP=GP-1E3:IQ=FNC(IQ+FNA(6)):IQ=IQ:GOSUB3380:GOTO2170
2200 IF0$<"N" THEN PRINTY$:GOTO2180
2210 IFGP<1000 THEN620
2220 Z$="DEXTERITY":GOSUB3370:IF0$<"Y" THEN2240
2230 GP=GP-1E3:DX=FNC(DX+FNA(6)):Q=DX:GOSUB3380:GOTO2210
2240 IF0$<"N" THEN PRINTY$:GOTO2220
2250 IF(GP<1000)OR(LF=1) THEN620
2260 PRINT:PRINT"WANT A LAMP FOR 1000 GP'S ";:GOSUB3290:IF0$<"Y
" THEN2280
2270 GP=GP-1000:LF=1:PRINT:PRINT"IT'S GUARANTEED TO OUTLIVE YOU!"
;GOTO620
2280 IF0$<"N" THEN PRINTY$:GOTO2260
2290 GOTO620
2300 Q1=1+INT(A/2):Q2=A+2:Q3=1
2310 IF(C1+4)>(T1)OR(BL=1)OR(DX<FNA(9)+FNA(9)) THEN2690
2320 PRINT:PRINT:PRINT"YOU'RE FACING ";C$(A+12):PRINT:PRINT"YOU
MAY ATTACK OR RETREAT";
2330 IFQ3=1 THEN PRINT", OR BRIBE";:IFIQ>14 THEN PRINT", OR CAST A S
PELL";
2340 PRINT:PRINT:PRINT"YOUR STRENGTH IS ";ST;" AND DEXTERITY IS
";DX:PRINT:GOSUB3280
2350 IF0$<"A" THEN2480
2360 IFW=0 THEN PRINT:PRINT"** FOUNDED ON ";C$(12+A);" WON'T HUR
T IT":GOTO2690
2370 IFB=1 THEN PRINT:PRINT"** YOU CAN'T BEAT IT TO DEATH WITH A
BOOK":GOTO2690
2380 IFDX<FNA(20)+(BL*3) THEN PRINT:PRINT" DRAT! MISSED":GOTO2690
2390 PRINT:PRINT" YOU HIT THE LOUSY ";RIGHT$(C$(A+12),LEN(C$(A+
12))-2):Q2=Q2-WV
2400 IF(A=9)OR(A=12) AND FNA(8)=1 THEN PRINT:PRINT"OH NO! YOUR ";W$(WV
+1);" BROKE":WV=0
2410 IFQ2>0 THEN2690
2420 PRINT:MC=MC-1:PRINTC$(A+12);" LIES DEAD AT YOUR FEET":IFH>1
-60 THEN2440
2430 PRINT:PRINT"YOU SPEND AN HOUR EATING ";C$(A+12);E$(FNA(8));
H=T
2440 IFX<R(1)ORX<R(2)ORX<R(3) THEN ON1-(A=13)GOTO2460,3220
2450 PRINT:PRINT"GREAT ZOT! YOU'VE FOUND THE RUNESTAFF":R(1)=0:R
F=1
2460 Q=FNA(1000):PRINT:PRINT"YOU NOW GET HIS HOARD OF ";Q;" GP'S
";
2470 GP=GP+Q:GOTO1420
2480 IF0$="R" THEN2690
2490 IF0$<"C" THEN2610
2500 IFIQ<150RQ3>1 THEN PRINT:PRINT"** YOU CAN'T CAST A SPELL NOW!
";GOTO2320
2510 PRINT:PRINT"WHICH SPELL (WEB, FIREBALL, OR DEATHSPELL) ";:IG
OSUB3290:PRINT
2520 IF0$<"W" THEN2540
2530 ST-ST-1:WC=FNA(8)+1:ON1-(ST<1)GOTO2690,2840
2540 IF0$<"F" THEN2580
2550 Q=FNA(7)+FNA(7):ST-ST-1:IQ=IQ-1:IF(IQ<1)OR(ST<1) THEN2840
2560 PRINT" IT DOES ";Q;" POINTS OF DAMAGE,";PRINT
2570 Q2=Q2-Q:GOTO2410
2580 IF0$<"D" THEN PRINT:PRINT"** TRY ONE OF THE OPTIONS GIVEN":IG
OTO2320
2590 PRINT"DEATH - - ";:IFIQ<15+FNA(4) THEN PRINT"YOURS":IQ=0:GO
TO2840
2600 PRINT"HIS:PRINT:Q2=0:GOTO2420
2610 IF0$<"B"ORQ3>1 THEN PRINT:PRINT"** CHOOSE ONE OF THE LISTED
OPTIONS":GOTO2320
2620 IFTC=0 THEN PRINT:PRINT"ALL I WANT IS YOUR LIFE!":GOTO2690
2630 Q=FNA(8):IFT(Q)=0 THEN2630
2640 PRINT:PRINT"I WANT ";C$(Q+25);", WILL YOU GIVE IT TO ME ";
2650 GOSUB3290:IF0$="N" THEN2690
2660 IF0$<"Y" THEN PRINTY$:GOTO2640
2670 T(Q)=0:TC=TC-1:PRINT:PRINT"OK, JUST DON'T TELL ANYONE"
2680 VF=VF+(PEEK(FN(XZ))=25):GOTO620
2690 Q3=2:IFWC>0 THEN WC=1:IFWC=0 THEN PRINT:PRINT"THE WEB JUST B
ROKE!"
2700 Z$=RIGHT$(C$(12+A),LEN(C$(12+A))-2):IFWC=0 THEN2720
2710 PRINT:PRINT"THE ";Z$;" IS STUCK AND CAN'T ATTACK":GOTO2750
2720 PRINT:PRINT"THE ";Z$;" ATTACKS ";IFDX<FNA(7)+FNA(7)+B
L*3 THEN2740
2730 PRINT:PRINT" HAH! HE MISSED YOU":GOTO2750
2740 PRINT:PRINT" OUCH! HE HIT YOU":Q=Q1:GOSUB2800:IFST<1 THEN28
40

```



```

6))
3240 PRINT"AN INTELLIGENCE POTION":IQ=FNC(IQ+FNA(6))
3250 PRINT"A DEXTERITY POTION":DX=FNC(DX+FNA(6)):IFLF=0THENPRINT
  "A LAMP":LF=1
3260 GOTO2460
3270 FORQ=1TO64:PRINT"*";NEXTQ:PRINT:PRINT:RETURN
3280 PRINT:PRINT"YOUR CHOICE ";
3290 INPUTO$:O$=LEFT$(O$,1):RETURN
3300 PRINT"HOW MANY POINTS DO YOU ADD TO ";Z$;:INPUTO$:PRINT
3310 Q=VAL(O$):IF(Q=0)AND(ASC(O$)<>48)THENQ=-1
3320 IF(Q<0)OR(Q>0)OR(Q<>INT(Q))THENPRINT"*" ";GOTO3300
3330 OT=OT-Q:RETURN
3340 INPUTO$:Q=INT(VAL(O$)):RETURN
3350 PRINT:PRINTZ$;:INPUTO$:Q=INT(VAL(O$)):IF(Q>0)AND(Q<9)THENKE
  TURN
3360 PRINT:PRINT"* TRY A NUMBER FROM 1 TO 8":GOTO3350
3370 PRINT:PRINT"WANT TO BUY A POTION OF ";Z$;" FOR 1000 GP'S ";
  :GOTO3290
3380 PRINT:PRINT"YOUR ";Z$;" IS NOW ";Q:RETURN
3390 PRINT"HERE IS A LIST OF ";Z$;" YOU CAN BUY (WITH COST IN <>
  ):PRINT:RETURN
3400 PRINT"YOU ARE AT (";V$;" ";Y$;" ) LEVEL ";Z:RETURN
  READY

```

# The End

## WHAT DO THEY SAY?

What do the reviewers say about  
SWORDQUEST and DEATH PLANET?

### SWORDQUEST for the PET

"For those who want the very best."

— Recreational Computing

"The only fantasy simulation I have seen with graphics and animation."

— Pet Gazette

"Challenging and Exciting"

— Kilobaud Microcomputing

"The SWORDQUEST fever is once again at its peak."

— R. Zamora, Editor, Recreational Computing

You seek the jewels and holy sword in the tomb of a long-dead wizard.

SWORDQUEST cassette and instruction booklet ..... \$12.95

### ESCAPE from the DEATH PLANET (with SOUND) for the PET

"Exciting Game"

— Popular Mechanics

"Fantasy Games Software has done it again... an exciting action game with good graphics, sound effects and real time programming. Graphics include animated laser fire and explosions."

— Kilobaud Microcomputing

Killer Robots and Imperial Stormtroopers search for you. Laser pistol combat inside a giant space station.

DEATH PLANET cassette and instruction booklet ..... \$12.95

Programs will run on any 8K or larger PET. Each cassette has both OLD ROM and NEW ROM versions.



**Fantasy Games Software**

P.O. Box 1683  
Madison, Wisconsin 53701



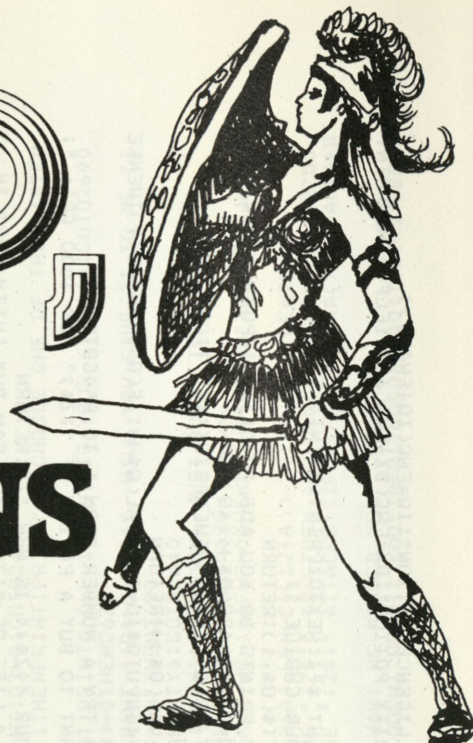
```

2750 IFQ<<"R"THEN3230
2760 PRINT"YOU HAVE ESCAPED":PRINT
2770 PRINT"DO YOU GO NORTH, SOUTH, EAST, OR WEST ";:GOSUB3290
2780 IF(Q$="N")OR(Q$="S")OR(Q$="E")OR(Q$="W")THEN 950
2790 PRINT:PRINT"* DON'T PRESS YOUR LUCK ";R$(RC):PRINT:GOTO277
  0
2800 IFQV=0THEN2830
2810 Q=Q-AV:AH=AH-AV:IFQ<0THENAH=AH-Q:Q=0
2820 IFQV<0THENAH=0:AV=0:PRINT:PRINT"YOUR ARMOR IS DESTROYED - G
  OOD LUCK"
2830 ST=ST-Q:RETURN
2840 FORQ=1TO750:NEXTQ:PRINTCHR$(12):GOSUB3270
2850 PRINT"A NOBLE EFFORT, OH FORMERLY LIVING ";R$(RC):PRINT
2860 PRINT"YOU DIED FROM A LACK OF ";:IFST<1THENPRINT"STRENGTH"
2870 IFQ<1THENPRINT"INTELLIGENCE"
2880 IFQ<1THENPRINT"DEXTERITY"
2890 PRINT:PRINT:Q3=1:PRINT"WHEN YOU DIED YOU HAD: ";:PRINT:GOTO29
  70
2900 Q3=0:PRINT:PRINT"YOU LEFT THE CASTLE WITH":IFQV=0THENPRINT
  "OUT";
2910 PRINT"THE ORB OF ZOT":PRINT:IFQV=0THEN2940
2920 PRINT:PRINT"A GLORIOUS VICTORY!";:PRINT
2930 PRINT"YOU ALSO GOT OUT WITH THE FOLLOWING: ";:PRINT:GOTO2960
2940 PRINT:PRINT"A LESS THAN AWE-INSPIRING DEFEAT."
2950 PRINT:PRINT"WHEN YOU LEFT THE CASTLE YOU HAD: ";:PRINT
2960 IFQ3=0THENPRINT"YOUR MISERABLE LIFE"
2970 FORQ=1TO8:IF(Q)=1THENPRINTC$(Q+25)
2980 NEXTQ:PRINTW$(WV+1):PRINTW$(AV+5):IFLF=1THENPRINT"A LAMP"
2990 PRINTFL$:" FLARES":PRINTCP$:" GP'S":IFRF=1THENPRINT"THE RUNES
  TAFF"
3000 PRINT:PRINT"AND IT TOOK YOU ";T$;" TURNS!";:PRINT
3010 PRINT:PRINT:PRINT"PLAY AGAIN ";:GOSUB3290:PRINT
3020 IFQ$="Y"THENPRINT"SOME ";R$(RC):;"S NEVER LEARN":PRINT:PRINT
  :GOTO 80
3030 IFQ<<"N"THENPRINTY$:GOTO3010
3040 PRINT"MAYBE DUMB ";R$(RC):;" NOT SO DUMB AFTER ALL":PRINT:EN
  D
3050 PRINT:PRINT"GREAT UNMITIGATED ZOT!";:PRINT
3060 PRINT"YOU JUST FOUND THE ORB OF ZOT!";:PRINT
3070 PRINT"THE RUNESTAFF IS GONE":RF=0:OF=1:OK(1)=0:GOTO1420
3080 DATAAN EMPTY ROOM,,THE ENTRANCE,E,STAIRS GOING UP,U
3090 DATASTAIRS GOING DOWN,D,A POOL,P,A CHEST,C,GOLD PIECES,G
3100 DATAFLARES,F,A WARP,W,A SINKHOLE,S,A CRYSTAL ORB,O
3110 DATAA BOOK,B,A KOBOLD,M,AN ORC,M,A WOLF,M,A GOBLIN,M,AN OGR
  E,M
3120 DATAA TROLL,M,A BEAR,M,A MINOTAUR,M,A GARGOYLE,M,A CHIMERA,
  M
3130 DATAA BALROG,M,A DRAGON,M,A VENDOR,V,THE RUBY RED,T
3140 DATATHE NORN STONE,T,THE PALE PEARL,T,THE OPAL EYE,T
3150 DATATHE GREEN GEM,T,THE BLUE FLAME,T,THE PALANTIR,T,THE SIL
  MARIL
3160 DATAT,X$,"?",NO WEAPON,W,ICH
3170 DATADAGGER," STEW",MACE," SOUP",SWORD," BURGER",NO ARMOR,"
  ROAST"
3180 DATALEATHER," MUNCHY",CHAINMAIL," TACO",PLATE," PIE"
3190 DATAHOBBIT,ELF,MAN,DWARF,I,I,A,L,I,Z,I,N,"C,A
3200 X=FNA(8):Y=FNA(8):IFPEEK(FND(Z))<>1Q1THEN3200
3210 POKEFND(Z),Q:RETURN
3220 PRINT:PRINT"YOU GET ALL HIS WARES: ";:PRINT:PRINT"PLATE ARMOR
  ";AV=3:AH=21
3230 PRINT"A SWORD":WV=3:PRINT"A STRENGTH POTION":ST=FNC(ST+FNA(

```



# YOU, TOO, CAN FIGHT DRAGONS



BY JOHN M. MORRISON

We all know what computers were built for from their inception—to play games on.

It is an extremely unusual, and probably extremely dull as well, programmer who has not owned, run, or “modified” such programs as *Star Trek*, *Lunar Lander*, and a relative newcomer called *Adventure*. These games, particularly the last one, belong to the genre of Fantasy Role-Playing Games.

“A fantasy role-playing game is one where the players (sometimes one, often several) can assume the personas and personalities of imaginary beings—not always human—who are thus given lives and adventures of their own, directed by the game players.

In the course of the game, characters (the imaginary ones) are presented with circumstances created in outline by a Game Master. The fulfillment of that outline depends on a verbal give-and-take between the Game Master and the players as they explore the world presented, taking actions to cope with the situations in which they find themselves.

The Game Master is a real person who has taken the time to create the scenario in which his or her friends can play. The players may be using medieval knights, space-faring aliens, or a cowpuncher from the Old West.”

So, a game of this sort basically consists of some fictional characters, who are controlled by players, running around in a universe devised by a person who takes the place of both God and the senses of the adventurers. This person, the Game Master, must not only create a situation but must describe what the players see, hear, etc. as if they were there themselves. They’ve been described in other places as “complicated forms of ‘Let’s Pretend’.”

Is it any wonder, then, that the task of Game Master would eventually be handed to computers? After all, a computer is certainly more consistent and does not (often) forget rules. The first game of this sort, created by Don Woods and William Crowther, was *Adventure*. In this game, the player is given a description of a scene (e.g., “You are in a dry stream bed”) and is then asked what he or she wishes to do. Although it might seem tame, there are homicidal dwarves that pop up to kill the unwary, and fabulous treasures that must be gathered before one can be considered an “expert.”

After *Adventure* came *Zork*, which was written on the MIT DEC-10 but soon found its way onto literally hundreds of other computer systems. Essentially, *Zork* is similar to *Adventure*, but has more rooms, more complex puzzles to solve, and a better parser. (A “parser” is a routine that translates English, or other languages, to a format understandable by the machine. More on this later.) There are several others of its ilk (I have played *Haunted House* and *The Shire*, to name two), all similar in that they first describe a situation and then ask for input.

With better terminals came more games. On the PLATO terminals on many different computers are several such as *Dungeons, Monsters, and Treasure* and *Orthanc*, where a little figure is propelled through a series of corridors and rooms, and *Moria* and *Baguette* in which the player sees an actual visual representation of what the fictional character would see. In another, a joystick-controlled tour through an entire imaginary town may be taken—all, of course, in acceptable perspective.

But as any avid computer gamer in college knows, games stand very little chance of staying on-line. Administrators have an odd aversion to them. Thus... the microcomputer.

Already many of the earlier programs, notably *Adventure*, are available for many of the micros on the market. Naturally, though the advancement of the art did not stop there. Innovative folks created so many role-playing game programs that they are impossible to list here, although a good overview can be obtained by surveying advertisements in personal computer magazines.

It is the aim of many to create their own programs. Sometimes a little help is needed to get started in such projects—thus, this article.

It really doesn’t matter what the adventure itself is. Although not a good tactic in writing fiction, a game set in the Old West can be changed to a one in the Far Future by changing “six-gun” to “proton blaster,” etc. The manipulation of data is the most important part. A good FRP program should be able to deal with any information it is presented with. Of course, any program should have this characteristic, but role-playing games have special problems. For example, a rather common spell in the *Dungeons and Dragons* rule system is “Charm

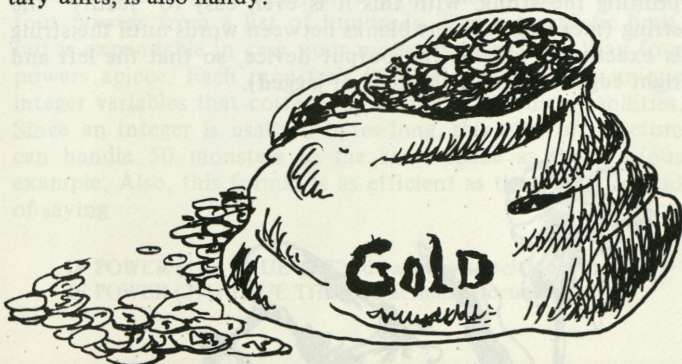


Person," which can bring a human under the control of another. Any competent Game Master can determine whether or not the spell is successful and the extent to which the victim is controlled. But how does a computer translate these parameters into ones that it can understand?

Let's start with a simpler problem: what must be in any FRP system? Naturally, one or more *characters* are necessary; because of the way terminals are set up (one keyboard to one screen), it is easiest to implement one-character games. More players, especially with autonomy of their own, cause both the economics and the intricacy of the game to increase geometrically. There are new software packages on the market, though, that allow one to tap into larger computers, so the outlook for multiplayer games is not hopeless.

These characters need certain *abilities* to flesh them out. These may take the form of Strength, Intelligence, Dexterity, Beauty, Luck, Experience, various levels of skills, etc. Most games operate with two to eight of these characteristics, but because they can easily be put into integer variables, a larger number can be implemented on a computer. Don't, however, fall victim to the other extreme—that of having too many statistics. If REFLEX-SPEED and MANUAL-DEXTERITY are used to refer to the same thing, one can be deleted.

An *inventory* is another important data field. Characters will not be fighting their foes bare-handed all of the time (although you may wish to start them out that way); swords, guns, shields, etc., should be on sale someplace or findable within the adventure. If you allocate an array, or a number of arrays, you can easily work with any particular item. Although this limits characters to a certain number of possessions, I can really not imagine most people carrying more than twenty individual things. A more advanced method is to use pointers; in other words, the array of items would have an additional field, NEXT. The player would have a variable indicating the number of this first item, and the next item would be the number in NEXT of the previous item. If NEXT is negative (or some other key value) that item is the last in the character's list. This method also lets items own other items such as a backpack or a duffel bag. In large dungeons this can save quite a lot of space. Gold, or other monetary units, is also handy.



The program deals with manipulating characteristics and possessions in various environments. Two types of program structure are evident at this point: "hard" and "soft" (as in hardware and software).

Adventure is an example of the hard-type. The room descriptions are enclosed within the statements of the program. A more extreme example would be one where, every so often (a hundred lines or so) a new room starts and is self-contained within a block. This happens most often when one writes programs at the terminal, thinking them out as one goes along. Except for very small, very restricted varieties, it is not recommended.

Of course, there are exceptions to all rules. It seems that if you can take adventures and break them down into sub-units (for example, in one named DRAGONQUEST the first subadventure may be ROOM-1, which is further broken down into TAKE-SWORD, READ-BOOK, and EXIT-TO-EAST), you can run them using a simple recursive procedure that might go



```
ADVENTURE (name):
INPUT COMMAND
IF COMMAND = option #1 THEN ADVENTURE (sub-
  adventure #1)
IF COMMAND = option #2 THEN ADVENTURE (sub-
  adventure #2)
*
*
list of other options
*
etc.
```

The list of commands, of course, would include basic ones other than "options" (this idea of the "command loop" will be covered later). Once a command is executed, the next higher level of recursion would take over. This is a useful tool when a subadventure can take place in a number of locations, for example, a book that can be read anywhere.

The soft type has many advantages, one of which is that by changing the data a totally new adventure can be created (especially if read in from a file). In fact, you may have to write an auxiliary program to input the data. Of course, watch who you give the "editor" program to, as anyone who has it can easily cheat in the actual play of the adventure!

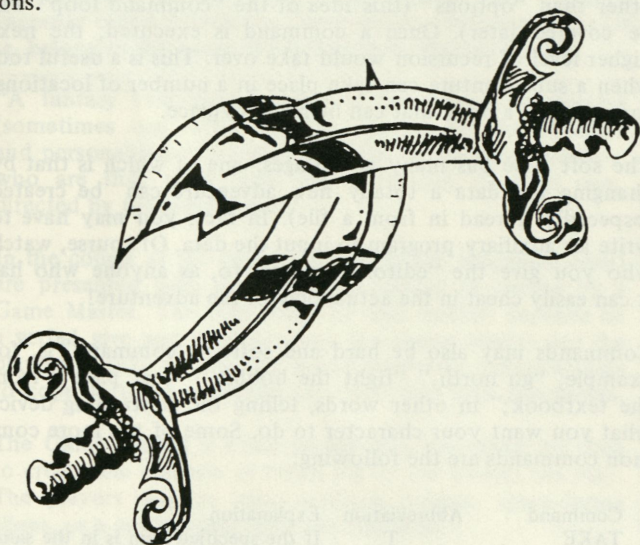
Commands may also be hard and soft. A 'command' is, for example, "go north," "fight the hobbit," "tear page 7 from the textbook;" in other words, telling the controlling device what you want your character to do. Some of the more common commands are the following:

Command	Abbreviation	Explanation
TAKE	T	If the specified item is in the same location as the adventurer, remove it from the room's possessions and place it into the character's.
DROP	D	Reverse of Take.
PRAY or CAST (spell)	P/C	Activates magic inherent in object, area, or character.
LOOK	L	Repeats description of area.
USE	U	May activate special feature of an item.
FIGHT	F	Engages foe, if present, in combat.
INVENTORY	I	Lists items character has.
EXAMINE	X	Takes a closer look at something.
Compass points (N, E, S, W, etc.)		Moves adventurer from one location to another if a connection exists in a specified direction.
QUIT	Q	Terminates game. If adventure is to be continued, may save data into a file.



The abbreviations are used in systems where string-handling is difficult or limited, or where a single character is needed, like in a realtime game.

The hard type is enclosed in the program; for example, typing "FIGHT" causes it to branch to a certain location or call a subroutine. In some areas this is wasteful, as in a room where there is nothing to fight. The soft method offers many advantages; however, it is restrictive in some ways. Included in the latest description of an area, together with what the character sees and a list of items which are within reach, is a list of actions that the character can perform. Inputting the number or letter of the choice causes an event to take place; for example, "PRINT 'There is a bright flash of light. Your sword is glowing.'" and "ADD 1 to SWORD-ATTACK." Still another method is to combine the types, have options that are standard, as well as some that vary from room to room. (For a better explanation of this principle, see issue #5 of *Sorcerer's Apprentice* magazine.) The two can be combined in yet another way: write (in a file) several lists of pseudo-commands that are executed whenever the specific command is typed. For example, 'TAKE' would ask for the item to take, check the contents in the room, and either add it to the character's possessions if possible or print an error message if it was not. When a character's strength is increased, the "weight limit" might be increased as well. With this method, commands could be more easily enabled, disabled, and changed; FIGHT would only mean something where there was something to fight. Items with special abilities could be "switched off" or even changed completely in certain situations.



With the latter method, commands can be deleted and added within the framework of the program. For example, if you destroy an idol in a certain room, it is only logical that all references to that idol be removed. You might write an "erase option" pseudocommand to include in the command sequence for options like these. Perhaps the easiest way to do the description part of this is to put in in "phrases"—that is, units that can be replaced as the need arises. In a particular room, there may be 6 of these:

1. "This is a large sized room "
2. "with a large, jade idol in one corner and "
3. "with exits to the "
4. "east, "
5. "west, and south. "
6. "A ladder also leads up. "

Thus, is one were to take the idol, phrase 2 would be deleted. Smashing it would cause the phrase to be replaced by "with a pile of green fragments in one corner and ." It also seems that you can take the ladder, replacing phrase 6 with "There is a hole in the ceiling, out of your reach." And, of course, the east door is lockable from the other side; doing this will change phrase 4 to "east ( although this door is locked)."



Null strings are ignored. With this system you must be careful concerning punctuation and grammar; deleting "east" from "Doors lead east and south" produces "Doors lead and south." If you have the inclination, you might replace "door" with "doors" depending on how many exits there are. What is more important is a "formatter," a routine to make sure that a word is not broken at the end of a line and continued on the next line. One way to do this is to take a dummy string as many characters wide as your screen or printer and put words into it until it overflows, then backtracking one word and printing the string. With this it is even easy to "justify" the string (meaning to insert blanks between words until the string is exactly as wide as the output device, so that the left and right edges are straight instead of jagged).





Getting back to the hard command method, unless you wish to have the player type one word at a time, a "parser" is necessary. One of the more basic types can take two words and act upon them, as long as the first is a verb and the second a noun ("TAKE TREASURE," "FIGHT DWARF"). A more complex type, namely the one in Zork, first scans the input string to see if it knows all the words, then separates the sentence into clauses (separated by the word "and"). Each of these clauses has one noun (the object) which has an action done to it and possibly one noun (the actor) which does the action, as well as a verb (the action itself). The matter is complicated with adjectives and modifiers; the result is that the parser can understand such statements as "PUT THE RED BOOK ON THE TABLE." Here the actor is the character. "TABLE" is an item that can possess items, if they are given to it in the right way. "BOOK" is the object of the action, and "RED" is an adjective that is referenced in case the character has more than one book. All "THEs" are ignored. "PUT," of course, is the action, which removes the red book from the character's possessions and puts it into the table's possessions. This principle dates back to T. Winograd's work in robot control and *SHRDLU*. (See *Understanding Natural Language* and the July-August issue of this magazine for more information.)

A problem, similar to the "Charm Person" example, now rears its ugly head in many a would-be program—Monsters (the term refers to anything hostile in the dungeon). To keep them from being mere faceless attack and defense values, special abilities are often included. To cite a well-known example, werewolves may only be killed with silver weaponry. One way to implement this is to have a Boolean array called "POWERS," in which POWER(1) would be the ability to take nonsilver damage without dying. This soon grows out of proportion: let's say you had twenty special abilities (actually a small number . . . the imaginative Game Master usually has hundreds if not thousands) and twenty monsters (again, a piddling value). Since a byte is the smallest accessible unit of memory, the table of monster powers is 400 bytes large. This number grows quickly as monsters or powers are added, and may well outdo your computer's memory sooner than you wish.

A pointer structure is again the answer. Each monster can have four powers from a list of hundreds. This is a realistic limit, but is expandable in case your monsters have more than four powers apiece. Each monster's POWER array is four unique integer variables that correspond to the master list of abilities. Since an integer is usually 2 bytes long, this type of structure can handle 50 monsters in the same space as the previous example. Also, this format is as efficient as the other. Instead of saying

```
IF POWER (1) is TRUE THEN do actions associated with it
IF POWER (2) is TRUE THEN do actions associated with it
```

```
*
*
etc.
```

a simple loop might be necessary, as

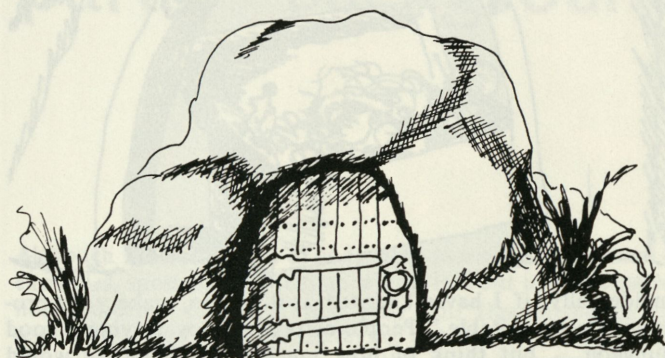
```
FOR I = 1 TO 4 DO
  IF POWER (I) = 1 THEN do actions associated with it
  IF POWER (I) = 2 THEN do actions associated with it
*
*
etc.
```

This might be the solution to putting spells and other magic in. Of course, the main problem—that of "would this item,

spell, etc. work in this situation?"—still holds. At present it seems that each case must be decided individually.

Once you have both a good system and a good program for using it, you will need a good adventure to implement. This, of course, is up to you and your skills as a writer and designer. However, there are some general pointers that are useful:

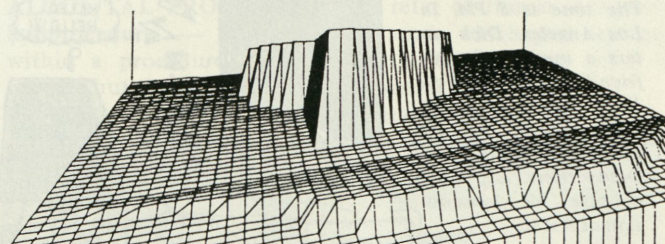
First, have some goal. A good one is getting out from a dungeon that one is trapped in. More common is the accumulation of gold or experience to a certain level.



Second, have subsidiary adventures along the way. These could very well yield helpful information or items.

Third, design it to be run from scratch. Not everyone may know that dimethyl ascorbate must be oxidized before it is edible, and if you have some in your adventure, many people will miss out on it unless you include a clue somewhere.

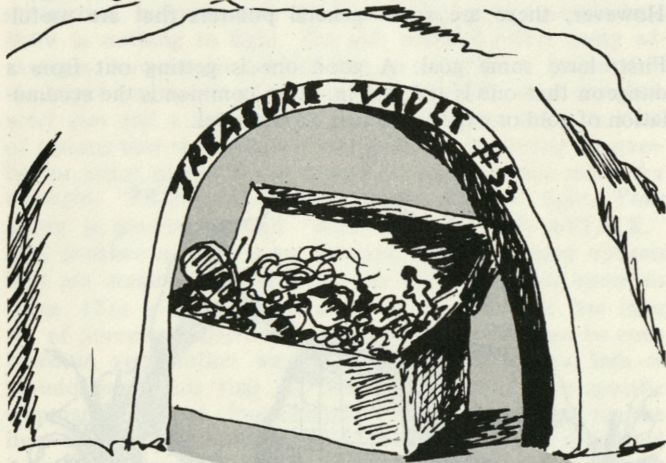
Peripherals and special features are tempting to include in a game design. Even with a joystick one can do a lot. However, remember that not everyone has a joystick—or music board, voder, or whatever. Games designed for the mass market should not contain these features. Random-access files, though, are fair game as disk drives are becoming so commonplace.



Still, they are fun to play around with. The value of some such as vocals (both for input and output) speak for themselves. (Sorry.) "Taps" played when a character dies is also hard to resist. Graphics are a perennial favorite, but the state of the art (without getting into the hundreds of kilodollars) is still not of photographic quality, which must be achieved before one can have a truly graphics-oriented fantasy game. Still, what is available is sufficient to draw inventory items and other helpful stuff. And with a bit pad or a touch-sensitive screen, you can not only manipulate items by hand but also explore on a map that is drawn on the screen or tablet. The example springs to mind of a corridor that has a trap in a certain section that must be circumnavigated—or else. Or how about a game where one could wield a light pen like a broadsword? The possibilities are limited only by how far your wallet happens to stretch!



After the adventure is complete, check it. It is a piece of software just like the usual type and is just as prone to errors (if not more so). You might find that the corridor that was supposed to lead to Room #35, the Ogre room, actually goes to #53, the Treasure vault. Have a friend check it out, and watch his game.



Incidentally, if I haven't hinted at it before, make your programs "user-writable." People certainly enjoy playing a good dungeon... but think of how much more fun they would have if there are hundreds of additional adventures (different in scope, yet bound under the same set of rules) floating around, some of which were composed by friends (or enemies)? Of course, you must remember the rule that large software firms seem to have forgotten: make sure it works before releasing it. If people are going to be plugging data into a program, make sure they can't plug the wrong data in, or at least tell them when they do. "Common sense," you say? You'd be surprised.

And a final, stern word of advice (although this article might already contain too many): before you hand it out or sell it or whatever, ask your self "could I make it better?" as a totally unrelated example, television reception would be ten times what it is today if industry had standardized not the first system that came along, but the best.

Enough reading. Get to work... you've got a universe to build. It's a big job, but somebody has to do it!

## BIBLIOGRAPHY

The following fantasy role playing games are all recommended:

*Tunnels & Trolls* and *Sorcerer's Apprentice* magazine, available from Flying Buffalo Inc., PO Box 1467, Scottsdale, AZ 85252  
*Traveller* and *The Journal of the Traveller's Aid Society*, available from Game Designer's Workshop, Box 432, Normal, IL 61761  
*In The Labyrinth* and *The Space Gamer* magazine, available from Metagaming, Box 15346, Austin, TX 78761  
*Dungeons & Dragons* and *The Dragon* magazine, available from TSR, PO Box 756, Lake Geneva, WI 53147  
*Runequest* and *Wyrms' Footnotes* magazine, available from The Chaosium, Box 6302, Albany, CA 94706

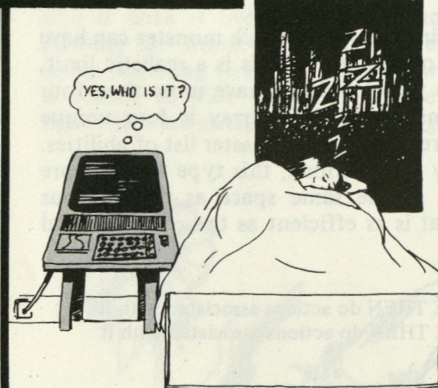
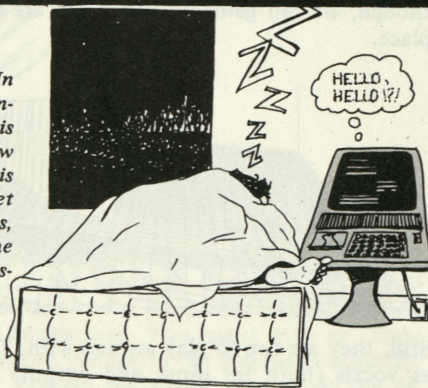
The following books may also be helpful:

*The Compleat Strategyst* by J. Williams  
 "Analyzing English syntax with a pattern-learning parser" by K. McConlugue and R.F. Simmons, in *Communications of the Assoc. for Computing Machinery*, vol. 8, no. 11  
*Understanding Natural Language* by T. Winograd  
*Introduction to Artificial Intelligence* by P. Jackson

In addition, the author would like to discuss problems and discoveries in the computer FRP games field, as well as games in general. Write to LPO Box 12613, Livingston C. Rutgers U., New Brunswick, NJ 09803.

# PCNET

The time is 8 PM. In Los Angeles, Dick enters a message for his friend Jane in New York City onto his PAN system. To get low night phone rates, Dick "time lags" the message for transmission at 3 AM.



3AM: Dick in Los Angeles and Jane in New York City are both asleep. Dick's PAN dials Jane's number, verifies that Jane's PAN is ready, and transmits the message. Both PANs then hang up the phones. Dick's PAN shows the message as "sent." Jane's PAN holds the message for review the next day on Jane's cassette tape.

PCNET, a project of People's Computer Company, has available a computer mail support software for the Commodore PET. Other versions for the Apple will be available soon.

The new PCNET computer mail system is called PAN—a program on

cassette tape for use with an 8K or larger PET. All that is required is a telephone line, an auto dial-auto answer modem, and a personal computer (available at present only for the Commodore PET).

The PAN software, a perpetual license for its use and a user's manual sell for

\$12; a user's manual is available separately for \$2.

If you would like more information on the PCNET project, or would like to order the PAN software including a perpetual license agreement, contact PEOPLE'S COMPUTER COMPANY, 1263 El Camino Real, Post Office Box E, Menlo Park, California 94025.



# STAR TREK—

## a dialogue approach

### part 1—background

BY SERG KOREM

*"Space, the final frontier . . ."*

#### INTRODUCTION

"... Let's see now, was fire torpedoes number 3? Or was that sensor scan?" How often have you played a Star Trek program and asked similar questions? Are you tired of fighting Klingons by "remote control"? Are you tired of fighting Klingons? Did the fun and thrill of playing that new Star Trek program warp away once you learned one game was pretty much like another? Have you become complacent with cruising through space for that matter? Or antimatter? If so, this series of articles may not only restore your initial excitement but hopefully it may teach you some simulation basics as well.

#### OBJECT

The intent of this series will be to lay out guidelines on how to build a Star Trek program "universe" which will recognize "conversational" as opposed to "numeric" commands, respond conversationally where appropriate, maintain a high level of interest and user interaction, and work on the smallest to largest of systems, single or multi-processor. The object of the program will be to achieve an output similar in form to the following:

KIRK: SCOTTY, GIVE ME MORE POWER IN THE PHASERS.  
SCOTT: AYE, CAPTAIN. FORWARD PHASERS NOW AT 151, AND REAR PHASERS AT 145 UNITS.  
KIRK: LOCK REAR PHASERS ON TARGET, MR. SULU.  
SULU: SETTING, SIR?  
KIRK: SETTING 85, MR. SULU.  
SULU: AYE, SIR. SETTING 85.  
KIRK: FIRE!  
SPOCK: HIT ON ROMULAN AT 2.6. SENSORS SHOW HE IS NOW AT 30.012 PERCENT CAPABILITY. ROMULAN AT 3.4 AT 15 PERCENT POWER.  
CHEKOV: ROMULAN AT 6.2 DESTROYED, SIR.  
SPOCK: CAPTAIN, 5 ROMULANS JUST DROPPED THEIR CLOAKING DEVICE.

(This was taken from an earlier version of my program.) When I say "conversationally," I mean gramatically correct, with the ability to be readily adaptable to speech synthesizers such as the Computalker.

#### APPROACH

Each article will deal with one aspect of the program, i.e., the ship, navigation, universe, etc. This should result in a highly modular program structure which will allow the individual programmer to choose which procedures he wishes to implement. A modular structure encourages expansion.

Although specific blocks of code will upon occasion be used as examples, most of the program will be laid out in a structured, non-dedicated language. Not only will this introduce the reader to a useful means of laying out program logic, it will also allow the user to implement the program in the language of his choice. Most of the specific code examples will be in BASIC although upon occasion I'll compare it to another language to point out BASIC strengths and weaknesses.

Even though I have written a similar program with great success, this will in essence be a totally new game. As such there may be, and probably are, ways of doing some things quicker, more efficiently, etc. Feel free to write to me (26 Marvin Dr. A4, Newark, DE 19713) to correct, suggest, or complain. Any suggestions used will be duly acknowledged in future articles.

A word about definitions. When I say

**Procedure** — I mean a block of code which processes data dealing with a particular function—navigation, sensors, etc. This is similar to BASIC GOSUB/RETURN pairs, and FORTRAN SUBROUTINES, but is closer in concept to ALGOL/TAL PROCs. I may also refer to this as a module.

**Subprocedure** — a block of code which gets repeated within a procedure. Nothing like a subprocedure exists in BASIC, but the concept is useful nonetheless.

**Block** — a group of coded statements.



Printed with permission from Paramount Pictures Corporation, "Star Trek—The Motion Picture."



## THE PSEUDO-LANGUAGE

This is a tool used for defining logic flow without the need for flow charts. A program is first defined in this language and can then be coded in any programming language, APL, BASIC, etc. The first statement type is fairly straightforward:

Declaratives — any operation:

```
A = A + 1
B = SIN(A/2)
GOSUB 2300
RETURN
etc.
```

The other statement types are conditionals:

```
IF (test 1) THEN (block 1) ELSE
IF (test 2) THEN (block 2) ELSE
...
IF (test n) THEN (block n) ELSE
(block for else);
```

Note that all pseudo-language statements are terminated in semicolons. The quantities within ( ) are not program code, but English language descriptions. The IF THEN ELSE conditional tests (test "n") if this evaluates true, (block "n") is executed. If (test "n") evaluates false, then the next IF THEN pair is executed. The final (block for else) is executed if none of the prior IFs evaluate true.

It is legal to omit (block "n") and/or (block for else) if no action is desired. So

```
IF (x is greater than y) THEN ELSE
IF (x is less than y) THEN ELSE
(print x);
```

would output the value of "x" when  $x = y$ . Note that this is equivalent to IF (x is equal to y) THEN (print x);.

The statement

```
REPEAT (block) UNTIL (test);
```

will execute (block) until (test) evaluates true. It is implicit in the structure of the statement that (test) is evaluated at least once and must reset (test) to false. On the other hand,

```
WHILE (test) DO (block) END;
```

will execute (block) as long as test evaluates true; that is, (test) is evaluated first.

```
FOR (range) DO (block) END;
```

will execute (block) as long as (range) some value is within the range. (range) is usually specified as I = 1 TO 3 BY 1 or similar statements. This is merely a modified BASIC FOR statement.

```
(block)
```

is defined as being;

```
BEGIN (group of statements) END;
```

where (group of statements) can be a nested pseudo-language statement set.

!Comments are denoted by imbedding them within exclamation points!

This is all that is needed to structure a program. Readers may have noticed that there is no GOTO statement. Although they may be used, virtually all programs can be structured (although not necessarily coded) without them. In this manner program flow will tend to be top-down and will result in more readable code.

## INTERACTIONS

The first step in setting up a simulation is to define all of the procedures and how they will interact. This can be done in several ways—flow charts, descriptions, or block diagrams. Because we are dealing with general concepts at this point, we will use block diagrams followed by a brief description of each element. Figure 1 is the Interactions Block Diagram.

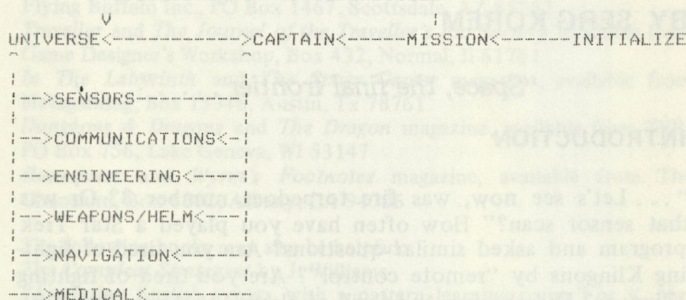


Figure 1 Interactions Block Diagram

**INITIALIZE** — Define and set all program variables. This includes randomizing the random number generator.

**MISSION** — Determines the goal of the game. This can be as simple as "destroy 76 Klingons in 32 days" or as difficult as hand to hand combat with a Gorn.

**CAPTAIN** — This is the central unit of the game and the person who will react to the situations created by the program — the use. Although this can be hardcoded as James T. Kirk (the captain on the show), it is usually the actual name of the player. More on this later.

**THE SHIP** — THE U.S.S. ENTERPRISE or whatever you prefer. **SENSORS** — Instruments which provide information about the universe to everyone aboard ship. **COMMUNICATIONS** — Allows the Enterprise to communicate with various and sundry ships, entities, etc. in the universe. The sensors and communications are the only sources of information about the universe the Captain has at his disposal, except for God or any other powerful being or force capable of bypassing the ship's systems. **ENGINEERING** — Maintains the ship's systems. Sort of a ship's utility company. **WEAPONS/HELM** — Just that. Allow the ship to defend or attack and maneuver within the universe. **NAVIGATION** — Tells the ship how to get "there" from "here." **MEDICAL** — Will play a minor role in this program, but can be expanded if desired.

**UNIVERSE** — All else. Other ships, starbases, planets, suns, life forms, etc.

## DIALOGUE

Since this is to be a dialogue-based game it would be perfectly valid to set up a procedure to handle this, but because the series is geared to systems of every size, it is better to dynamically generate conversions as needed in order to

Continued on pg. 26



## MARKETPLACE

Classified ad space available: \$60 per vertical inch.  
Columns are 3 1/4" wide.

### COMPUTER SUB HUNT

Track an enemy sub around an unknown ocean, get close and fire depth charges. Send \$7.95 plus \$1.00 shipping to SATALITE COMPUTER CO., 38 Bourne Avenue, Tiverton, RI 02878—order now—For the TRS-80—Level I.

### ComputerTown, USA!

A grass roots, economical model of how to offer everyone in a community of 27,000 the opportunity to use a microcomputer. Your interest and help is invited. For more information, please send a large self-addressed, stamped envelope to: ComputerTown, USA!, Box E, Menlo Park, CA 94025.

### PCNET PAN

An electronic mail package, PAN allows PET owners to send and receive messages over the telephone network. Entirely written in BASIC, PAN permits immediate message transmission, or unattended transmission at a specified time. PEOPLE'S COMPUTER COMPANY, PCNET Project, P. O. Box E, Menlo Park, CA 94025.

for TRS-80 II, Compucolor II,  
& PET



### EVERYTHING YOU NEED TO BRING YOUR BASIC PROGRAMS TO LIFE WITH SOUND!

SOUNDWARE is a complete package:  
**YOU GET A SPEAKER/AMPLIFIER UNIT** complete with connectors. No wiring or soldering. Just plug in!

**YOU GET A DEMO PROGRAM** with a variety of sample sound effects—sirens, laser sounds, tunes!

**YOU GET A COMPOSER PROGRAM** to help you create your own original sound effects. Fun for all ages!

**YOU GET AN INSTRUCTION BOOKLET** that tell you how to insert sound into your programs. 1 year warranty.

	Complete Package	Programs or & Booklet
ALL PETS & TRS-80 II	\$29.95	\$14.95
Compucolor II	\$39.95	\$19.95

SEND FOR FREE CATALOG OF  
GAMES FOR PET & COMPUCOLOR!!

SEE YOUR DEALER TODAY! Or order direct from CAP by phone or mail. VISA & MasterCard orders include expiration date. Add \$1 postage & handling per order, \$3 for air or COD. Arizona residents add tax.

CAP Electronics 8462 Hillwood Ln., Suite B, Tucson, AZ 85715 (602) 296-4978

# WILEY SELF-TEACHING GUIDES... Keeping up with the fast-moving world of recreational computing



From the Editors of *Recreational Computing*...

### TRS-80 BASIC

Bob Albrecht, Don Inman, & Ramon Zamora

Packed with games, graphics, and practical applications, this eagerly awaited guide leads you step by step to maximum use and enjoyment of your new TRS-80.

"The book is OUTSTANDING and is perfect for all LEVEL II owners. . . geared toward the beginner, but does contain a wealth of information for more advanced programmers. . . TRS-80 BASIC is the best thing to happen to LEVEL II owners!"

*Computronics Monthly News Magazine*

0 471 06466-1 1980  
351 pp. \$8.95 paper

### STRUCTURED COBOL

Ruth Ashley

Produce programs that are easier to read. . . less costly to debug. With this complete self-instructional guide to the computer language most widely used in business, you'll write or modify programs more swiftly and accurately than you ever thought possible.

0 471 05362-7 1980  
295 pp. \$8.95 paper

### JOHN WILEY & SONS, Inc.

605 Third Avenue  
New York, N.Y. 10016  
In Canada: 22 Worcester Road, Rexdale, Ontario  
Prices subject to change without notice.



1-7715

### BACKGROUND MATH FOR A COMPUTER WORLD, 2nd Ed.

Ruth Ashley in consultation with Nancy B. Stern

Here are all the basic mathematical techniques, facts, and concepts you need for productive, intelligent interaction with your computer using any computer language. The new edition of this popular manual includes an entirely new chapter on trigonometry.

0 471 08086-1 1980 304 pp. \$7.95 paper

These are the newest titles in Wiley's fast-growing line of popular STGs to help you make the most of your personal computer. Look for all of them at your favorite bookstore or computer store!



prevent overcoding by small system users. Users with discs and/or professional systems can experiment with modifying the program to work on key-sequenced files, sentence parsing, dictionaries, etc. These methods are an integral part of Artificial Intelligence, a rapidly growing computer frontier. Anyone attempting this and having any success (or problems) can contact me at the above address.

Dialogue implies people, so let's define all of the players in our universe.

(the user) — The captain of the ship.

CMDR. SPOCK — Science officer — in charge of sensors, ship's computer, etc. He is from the planet Vulcan, shows little if any emotion, and is highly logical. He is second in command.

LT. CMDR. SCOTT — Chief engineer — in charge of ship's systems and power. Will tend to talk in a Scottish brogue when excited.

LT. UHURA — Communications officer — female.

LT. SULU — Navigator — in charge of piloting the ship.

ENS. CHEKOV — Weapons/Helm officer — Occasionally takes over Spock's station. Speaks with a Russian accent.

DR. LEONARD "BONES" MCCOY — Chief medical officer — Captain's moral support and advisor.



Printed with permission from Paramount Pictures Corporation, "Star Trek — The Motion Picture."

ENGLISH	SCOTTISH	RUSSIAN
SIR		SER
VERY		WERY
CAPTAIN		KEPTIN
NOT	NAE	
CANNOT	KENNA	
DID NOT	DIDNA	
KNOW	KEN	

Figure 2 Dialect Equivalence Table

It is extremely important to keep these "players" in character. That is, Sulu must have very little if anything to do with communications. It is permissible, however, to have a person temporarily take over a station similar to this own. This will help to produce a feeling of reality in the game.

Now you may be saying to yourself: "But I don't know what a Scottish brogue sounds like." Well, fear not! Figure 2 is a "Dialect Equivalence Table" for Chekov and Scott.

Replace any of the words in the English column when used by either Scott or Chekov with the corresponding equivalent. In addition, Chekov pronounces all v's as if they were w's, and not the other way around — that would result in a German pronunciation. Also, Scott speaks with a brogue only when excited or worried. Since the Enterprise is a paramilitary vessel, realism is enhanced by maintaining a sense of military decorum; that is, acknowledge any commands with "Aye, sir," occasionally repeating the command.

The format of the output should be the following

```
(user): (sentence)
or
(character): (sentence)
```

This should result in a continuous stream of dialogue. Of course, it will still be necessary to display quadrants, sensor plots, etc. And too, a command from the user has to be imbedded in this dialogue (more on this in the future).

There are many novelizations of the program's episodes, and the programs themselves can yield information on how to keep your crew in character.

## A WORD ABOUT IMPLEMENTATION

A particular user's version of this program will by necessity be limited in scope by the size and type of system he has available. The critical limiting factor will be the total memory allocated to strings and their manipulation. As I've mentioned, a program of this nature should ideally be done on a large, multi-processor system with a lot of disc space and keyed-files. This should in no way dissuade the home system user from attempting to code the program since the techniques presented can be applied to many simulation situations.

The home system user will have to tailor the program to suit his system's capabilities; this will mean implementing from the necessary code out. Strings will have to be shortened or eliminated. The result, however, will still be more interesting and fun than other Star Trek programs.

To get the full benefit of the program it is suggested that small systems users program a procedure at a time until a memory overflow situation occurs. Then prune away code which you don't consider important until the program runs smoothly. Do not be afraid to improvise and improve. Users lucky enough to have large systems at their disposal can expense procedure capabilities, convert the program to multi-user operation, etc.

## NEXT TIME

In the next installment we will define and initialize all of the program variables, and if space permits begin work on the universe procedure. As an exercise, the user is encouraged to play god and consider how you would implement such a universe and what types of variables would be needed. After all, simulation is an attempt to bring to life by means of a computer that which may or may not exist.

*"live long and prosper"*



# ON FUTURE FANTASY GAMES

BY DENNIS ALLISON

## Introduction

Even the most complex of today's computer fantasy simulation games is simple-minded compared with what is possible. Limitations in concept and execution appear in many places. Most games are built around the simple "explore and collect" scenario. Interaction with the fantasy environment is limited by the computer hardware on the one hand and by the limitations of the software on the other.

## Pictures And Sound Would Help

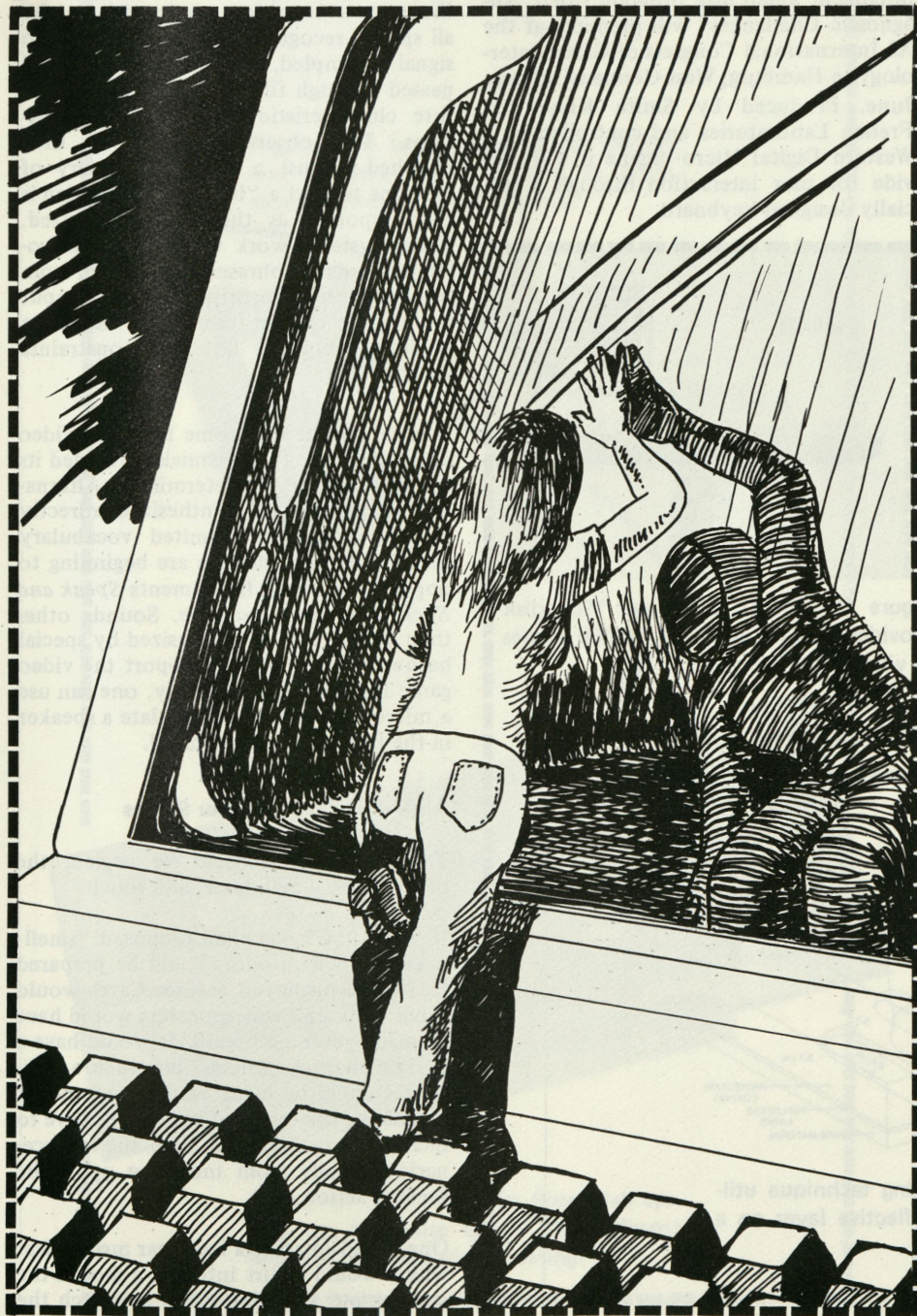
We humans depend upon our eyes and ears; they are our primary input devices. When the game tells us what to see we create the vision with our mind's eye. Literary devices aside, if we could see and hear what was happening the game would have the immediacy of a movie or TV program and less that of a book.

One thinks immediately of animation, drawing the events of the game out onto the computer terminal's screen. Some games make limited use of the computer's ability to present information graphically (maps, for example) but none provides true animation. The necessary resolution for an acceptable quality picture is not available nor is the processor power. It takes a lot of computation to make a single minute of animation.

Video tape might appear to be a solution. A control unit is available from Cavri Systems (26 Trumbull Street, New Haven, CT) which allows a computer to control a videotape machine. But tape is a linear medium; you can't switch around as a game demands.

Enter the video disk. It seems tailor-made for the gamesmith. Video disks have been designed primarily as movie delivery systems and high quality sound systems. But range of application will be much larger: digital data storage, computer-assisted instruction, and making computer fantasy games more real, to mention a but a few. U.S. Pioneer Electronics (85 Oxford Drive, Moonachie, NJ 07074) has recently begun marketing the video disk player shown in Figure 1 for a suggested retail price of \$750. It uses a recording technology developed by Phillips N. V. in Holland over the last ten years. While it is not designed for computer control, it is almost certain to be adapted for such use.

The primary advantage of a video disk is its ability to access any frame or sequence





of frames randomly. A standard disk can store 54,000 frames, each of which holds one screen picture. In normal sequential mode, that's enough to hold one half-hour segment of a movie. Digitally that's a lot of information, over 10,000,000,000 bits or some 1200 floppy disks. Frames located close together can be referenced with only minor interference with the video signal; the whole space from frame zero to frame 54,000 can be traversed in under 15 seconds.

The recording technology uses tiny pits in a reflective surface such as those shown schematically in Figure 2. The disk is scanned with a laser and the level of reflected light is measured to reconstruct the signal. The receiver mechanism is shown schematically in Figure 3.

Since the playback mechanism never contacts the disk, a single frame can be displayed continuously without any damage to the disk itself. Disks are relatively inexpensive to manufacture in quantity though the mastering process is complex; prices delivered to the consumer will be in the \$10 to \$20 range.

### Using A Video Disk In Fantasy Games

The idea is obvious: use the video disk to add pictures and sound to the game. That means careful planning because even a half-hour of video is short in a complex game; many scenes should be of the single frame variety.

One would like to be able to use animation and written output in conjunction with the output from the video disk. The video disk conforms to standard NTSC format for television signals. Unfortunately, most computer systems do not conform to that standard format because of a desire to produce higher resolution outputs. Of contemporary computers only the TI 99/4 and the Cromemco SDI board are truly NTSC compatible. The Cromemco board actually has the mixing facility on-board already.

Computer fantasy games are not the application for this mixture of computer output and recorded picture and sound. Already the same idea is finding application in computer-assisted instruction. An intelligent video disk program titled "Diagnostic Challenges" will premiere at the XI International Congress of Gastroenterology in Hamburg, West Germany in early June. Produced by Smith Kline and French Laboratories and controlled by a Western Digital Micro-engine it will provide for user interaction through a specially designed keyboard.

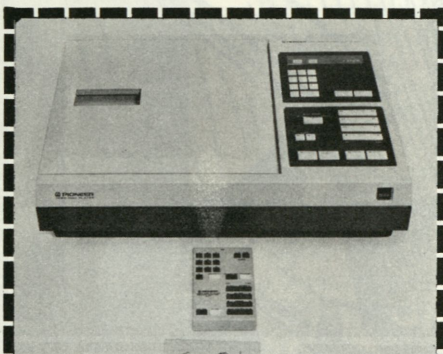


Figure 1. The U. S. Pioneer Videodisc provides random access to 54,000 frames of video information.

### VIDEODISC CONSTRUCTION

#### MAGNIFIED VIEW OF A VIDEODISC

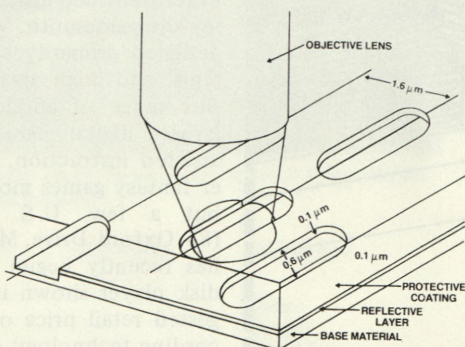


Figure 2. The recording technique utilizes small pits in a reflective layer on a plastic substrate.

### Sounds And Speech

Speech is a familiar and natural medium of communication. The terminal keyboard is not; in fact, many people who would like to play computer fantasy simulation games will not be able to type. This suggests other means of input would be desirable including touch screens and, most importantly, speech.

Speech recognition is a difficult problem. Several systems are on the market with single word or phrase recognition vocabularies of a few tens or hundreds of words after training. Figure 5 shows the Speechlab system for S-100 based systems produced by Heuristics (1285 Hammerwood Avenue, Sunnyvale, CA 94086). Much the same recognition process is used by all speech recognition systems. The audio signal is sampled, normalized in time, and passed through filters to develop a signature characteristic of word sound patterns. The observed pattern is then matched against a stored dictionary of patterns to find a "best" match, which is then reported as the word recognized. These systems work well only with isolated words or phrases but fail for continuous speech. Fortunately, speech input for a game can be trained and can use a small vocabulary. But the constraints are irritating.

Speech output can come from the video disk (there are two channels) provided its content can be predetermined. Alternatively, it can be synthesized directly by the computer. Limited vocabulary speech synthesis devices are beginning to appear; the Texas Instruments *Speak and Spell* is a good example. Sounds other than speech can be synthesized by special hardware developed to support the video game industry. Alternatively, one can use a microprocessor to manipulate a speaker in the fashion of the Apple II.

### Appealing To The Other Senses

There are five senses. We neglect the other three: smell, taste, and touch.

In the 1950's someone proposed "smell-o-vision" where odors could be prepared to fit the displayed action. Caves would smell dank and wet; monsters would have stomach-retching breath. It would have a limited market perhaps, but such a peripheral could be made easily; it would fit just above the terminal and would have to include a small fan to keep the last scenario's smells from intruding upon the present action.

One could imagine a headgear mechanism which would squirt into one's mouth the appropriate flavoring agent to match the



### PLAYBACK LASER PATH

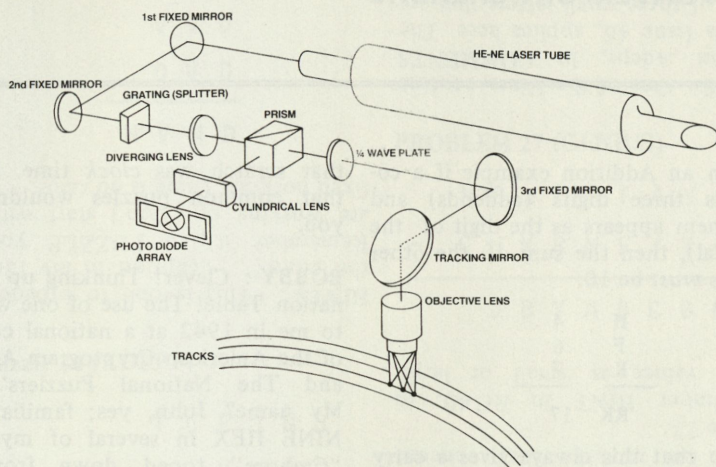


Figure 3. A laser is used to scan the disk and measure the reflectance to reconstruct the recorded video signal.

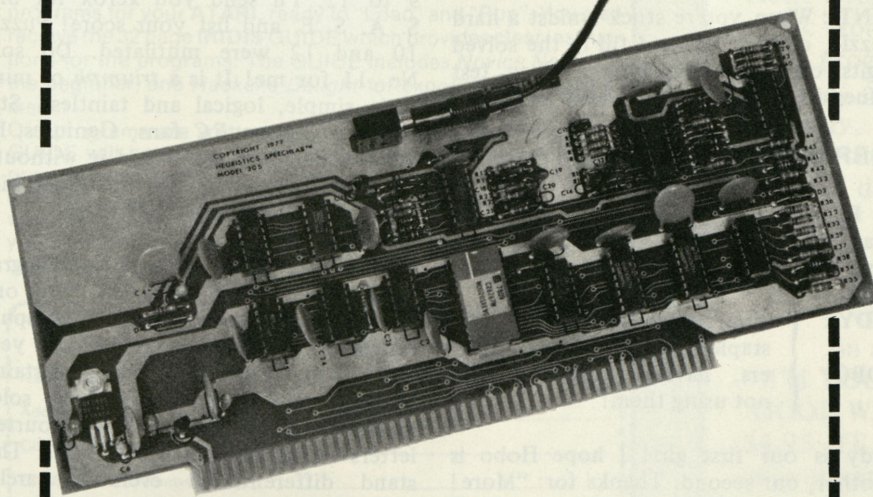
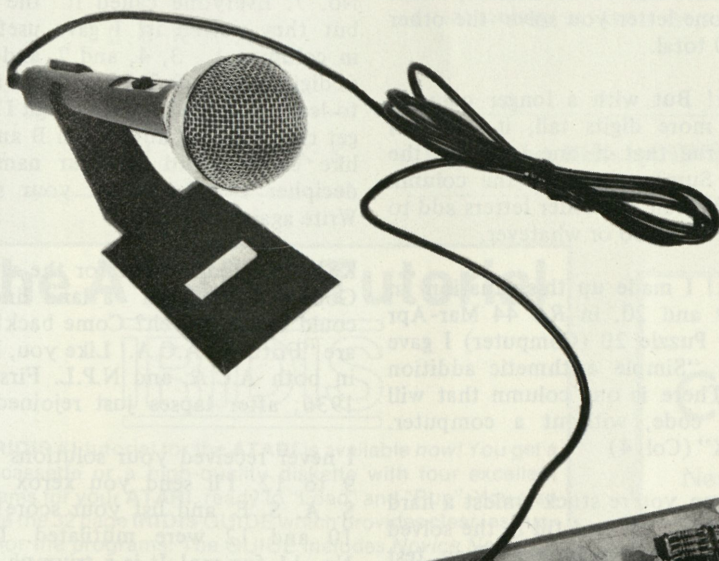


Figure 4. This Heuristics Speechlab system can recognize up to 250 different words after appropriate training.

story. A trip to *Candyland* might be quite an experience.

And then there's the possibility of feelies where the tactile senses are exercised. The computer would provide touch responses—the caress of a lover, the blow of a sword—all in response to the current scenario. Given the violence many adventure games seem to have, an effective implementation might actually be dangerous.

### Other Plots Are Possible

Even without all the fancy hardware, the limitations of today's fantasy games are primarily those of imagination and computation power.

The exploration paradigm has been popular because it's easy to implement and understand. But the imperative fast-moving adventure is only one of a universe of possibilities. In most games the interaction, albeit complicated, is pretty superficial. Games where the interaction is even more complex could be even more fascinating. Some different approaches were sketched in my *Epic Games* article published in the March-April 1978 issue of *People's Computers*. Consider, for example, the electric murder mystery. You, the player, might be cast in the role of the detective. Through the magic of video you might visit the scene of the crime, interview witnesses, collect evidence, and interact with a complex world. The plot and characters need not be fixed but could be developed from the program using built-in rules about people's motives, the physical act of committing a murder, and the myriad of other facts which go into developing a plausible plot for a mystery story.

Another kind of interaction might be the electric biography. Here the player would be able to interview the person and ask questions. Here the knowledge base would need to know about the personalities and events which shaped the person's life. Using this knowledge base the computer could select scenes and construct answers to the player's questions. Imagine being able to discuss palace politics with some group of courtiers from the court of Louis XIV!

Games of this complexity are not yet possible. But they are not too far from what is possible.

Many of the ideas here are supportable with techniques borrowed from artificial intelligence. The only real demands are large amounts of memory, large mass storage (the video disk might help here) and major quantities of computation power.



# CRYPTARITHMS

BY JOHN DAVENPORT CREHORE

Send your solutions to Jack Crehore,  
P.O. Box 96, Charlotte Court House,  
Virginia 23923

Here's a thrill! It is thrilling when you succeed in ferreting *arithmetic* Sequences out of code *letters*! The fun... uh-h... toil that I've had so far with Puzzle 27 persuades me that other near-Geniuses might enjoy the bit of educational exercise it offers.

I haven't solved it yet. I'm copying from my scratch papers, "6-6-41 Good problem; too hard for beginners."

So long as we're having fun, listen, you who keep urging me to use real words for keys instead of the 14 letters I now use. Real words give unfair opportunity to anagrammers to send in solutions without having done any mathematics at all.

Anyway Puzzle 27 (Genius), before I transposed it from your alphabet lingo into my 14 cryptic letters, Looked kind of cute to me at my advanced age of 89! I must have thought young in 1941, when I used the real alphabet. Here's 27: FOAMY DRINK and 8 little words under it in a subtraction example. Top line OKIFMANDRY. If I gave you more, you'd solve it.

**HINTS:** Start with *Sequences* found in Puzzle 27. The general procedure is simple. List every group of letters (digits) in which you can indicate actual differences in value or comparative differences in size—large or small.

Pick a letter, let's say C, and make a list of letters you have determined may be greater than C in digit value. When you find four digits larger than C, you know C *can't be larger than number 5*. This way you often can find values for several letters close enough to break an impasse. I find without my 1941 scratch:

H	>	C	Col. 10
A	>	C	Col. 4
R	>	C	Col. 7
Y	>	C	Col. 7

OK. That's my share! Rub your lamp and call your Genie to help you finish Puzzle 27. You'll need him!

**HINT:** In an Addition example if a column has three digits (addends) and one of them appears as the digit of the Sum (total), then the sum of the other two digits *must* be 10.

H	4
F	6
K	7
RK 17	

Note also that this always gives a carry to the next column of just 1 because with 10 for two letters, the third letter can't bring the total higher than 19. Note also that unless both letters are the same, neither letter can be 5. Besides, when you solve one letter you solve the other to make 10 total.

OK so far! But with a longer column, say 5 or more digits tall, it will still often be true that if one letter of the total (the Sum) appears in the column above it, then all the other letters add to a total of 10, 20, 30 or whatever.

Look back! I made up this situation in Puzzles 19 and 20, in *RC* 44 Mar-Apr 80. Under Puzzle 20 (Computer) I gave the Hints: "Simple arithmetic addition example. There is one column that will break the code, without a computer. NINE HEX" (Col. 4)

**HINT:** When you're stuck amidst a hard Puzzle, copy it off and fill in the solved digits close to their letters. Then test values at random for all nearby letters.

## CORRESPONDENCE

**S. R. McENTEE:** Beautiful work—so clean. Hard Puzzles don't fool you, I see. Thanks.

**JUDY:** { Your letters came to my desk stapled together. Keen solvers, having computers and not using them!

**HOBO:** { Judy is our first girl! I hope Hobo is another, our second. Thanks for: "More! More!" and "Enjoyed the puzzles!"

**BOB:** Headwork and computer too! You think fast if 20 minutes for all

that scratch was *clock* time. I can see that common puzzles wouldn't throw you.

**BOBBY:** Clever! Thinking up an Elimination Table! The use of one was taught to me in 1942 at a national convention of the American Cryptogram Association and The National Puzzlers' League. My name? John, yes; familiarly, Jack; NINE HEX in several of my statuses. "Crehore" toned down from feudal Irish "Crohoore" found in Plymouth colony in 1641.

**RICHARD:** Yes, two solutions to old No. 7. Everyone called it the hardest, but they solved it! I gave useful clues in columns 1, 3, 4, and 7, and fixed Y as digit 1, or 2, or 3. I liked it all too well to leave it unused even though I couldn't get rid of the dichotomy in B and R. I'd like clear record of your name; can't decipher two letters of your surname. Write again!

**K8VDU Fred:** Thnx for the wallpaper, Good Buddy. With a land line up we could modulate, eh? Come back! So you are "Fritz" in A.C.A.! Like you, I've been in both A.C.A. and N.P.L. First joined, 1936; after lapses just rejoined A.C.A.

I never received your solutions to Nos. 9 to 12. I'll send you xerox for only S. A. S. E, and list your score! Puzzles 10 and 12 were mutilated. Do solve No. 11 for me! It is a *triumph* of mine! It is simple, logical and taintless. Still, not one of our *RC* fans, Geniuses, has yet claimed to have solved it without a tip! It's Cancellation for 5th grade kids. Seventy-threes. Eights. NINE HEX.

**CHARLES:** Your Computer Program never reached me. Anyway, I'm only slightly qualified to attend to computer affairs. I quit giving word keys years ago. They make invalid any certainty that a contestant's solving be solely mathematical. I promote these fourteen letters—ABCFHJKMNRTUWY. They stand differentiated even in careless script. Do write again. Programmers must rank as experts! And we're all in this to learn, eh?

Jack Crehore—NINE HEX



### Puzzle 25 (NOVICE)

D V D  
V V D  
D V D  
N V N D

HINTS: "This is one to broaden the mind..." caption in my file. Remember it's a PUZZLE you are working on. Don't start with fixed notions. There is no error.

### Puzzle 26 (ADEPT)

T C F J N B  
Y T A C  
B B T Y M K  
M T C J Y B A  
C B K M A F  
N C C T Y N  
F J N T C B J K K

HINTS: One of the Hints, routine for an Adept, in COMMENTS in this Issue 46, applies here. The range of three small letters that are routinely tested due to position is detected at a glance. After that

### PROBLEM 27 (GENIUS)

H B J C R F A Y K M  
- C H K R M F J Y H B  
J B Y A B C B M M

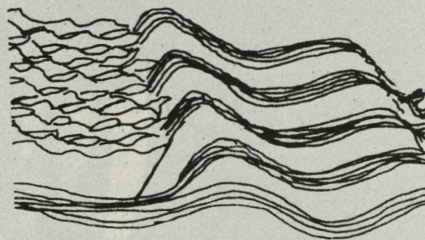
HINTS: A study in Sequences. I am sequencing clues amongst COMMENTS so that experts, by unintentional glimpses, do not lose the satisfaction they might have had from solving solo. Look for remarks that concern Puzzle 27. Remember HINT in RC40, to chart values as small or large.

Cryptarithm Hints by NINE HEX

### PROBLEM 28 (COMPUTER)

H U H H T R J  
M W H W T W J  
Y W H W T W  
M W H W T W J  
N W H H T W J  
B N H W H B J  
M T H U R H N W

HINTS: There is no Carry-to Column 5. With this knowledge you can fix T! and W!



## The ATARI® Tutorial

## IRIDIS

The IRIDIS #1 tutorial for the ATARI is available now! You get a C-30 cassette or a high-quality diskette with four excellent programs for your ATARI, ready to "Load" and "Run". You also receive the 32 page IRIDIS GUIDE which provides clear instructions for the programs. The GUIDE includes *Novice Notes* for the beginner, and *Hacker's Delight* for experienced programmers.

Our programs are written to be studied as well as used. The GUIDE will have complete source listings of selected IRIDIS programs. Not just listings, but an explanation of what's going on. If you are new to programming, IRIDIS is one of the easiest ways you can learn advanced techniques. If you're an old hand, you'll still find IRIDIS to be a rich source of ideas and ATARI techniques.

ATARI is a trademark of ATARI, Inc.

Please send me IRIDIS #1 for my ATARI immediately.

☐ \$9.95 Cassette (needs 16K) ☐ \$12.95 Disk (needs 24K)

Name \_\_\_\_\_

Address \_\_\_\_\_

City/State/Zip \_\_\_\_\_

☐ VISA Card Number \_\_\_\_\_

☐ MasterCard Expires \_\_\_\_\_

Published By:

Dealer  
Inquiries  
Invited

**The Code  
Works™**

Box 550  
Goleta, CA 93017  
805-967-0905

Programs for your ATARI®

## PET COMPUTER TEACHERS

New! Everything you always wanted in arithmetic software. The PET PROFESSOR is a 70 program package which teaches with a moving cursor technique... add, subtract, multiply, divide whole numbers, fractions and decimals. \$299 on cassettes or \$259 on 5 1/4" diskettes. Send \$5 for a sample tape.

Our STUDENT WORKBOOKS for beginning programmers has classwork, homework, hands on and off exercises. FEED ME, I'M YOUR PET (beginner) and LOOKING GOOD WITH YOUR PET (intermediate) are \$4.95. TEACHERS' PET (lesson plans, quizzes, answer key) is \$4.00



**COW BAY COMPUTING**  
BOX 515, MANHASSET, N.Y. 11030



THE





# WONDERFUL WORLD OF EAMON

BY DONALD BROWN

The troll smirks at you as you raise your puny sword and prepare for battle. His mood quickly changes when he sees a bright flame appear around your blade. The battle lust is upon you once more; you attack!

If you have never experienced this type of excitement in your life, you have missed one of the newest types of games around. This is a scene from a Fantasy Role-Playing game (FRP for short).

Unlike most games, there is no clear single goal in an FRP. Instead, you are directing the actions of a character to achieve the goals you want him/her to achieve. Although most people accept normal goals such as 'make lots of money' or 'become a feared warrior,' you can choose any yardstick you want to measure your success. After all, it's your life—you ARE the character. (For a more complete explanation of Role-playing games, see the May-June 1979 issue of *Recreational Computing*, "What Is ALL This Stuff?").

The first major FRP was *Dungeons and Dragons*, created by Gary Gygax and Dave Arneson. Since then many other sets of rules have appeared on the market: *Runequest*, *Tunnels and Trolls*, *The Fantasy Trip*, and others. They all work in roughly the same way; a special player called the *Dungeonmaster* or *Referee* beforehand secretly designs a building or cave complex and places monsters, traps, and treasure inside of it. Then, the other players join in the game and they direct their characters through the dungeon, receiving descriptions of what they see and describing what they do. In conjunction with the rules, the *Dungeonmaster* determines who survives the inevitable combat and how much loot they bring home.

It has occurred to some people that at least the second part of *Dungeonmastering* (the interaction with the players) is ideal for a computer, and thus they set forth to adapt the games for their own systems. Unfortunately, troubles arose. In addition to piddling questions of copyright infringement, the simple fact is that these systems do not adapt well to computers.

The first major problem is in combat. In these games, when Grimy the Gnome decides to swing at an orc (or vice versa), the *Dungeonmaster* has to consider a large number of factors and calculate the chance of Grimy actually hurting the little beggar. Since humans by and large do not enjoy doing half a dozen major additions or multiplications, the games usually use large tables to calculate chances to hit. Unfortunately, while the computer has no objection whatsoever to heavy arithmetic, storing tables uses up definitely limited memory.

Even more of a problem, however, is magic. Since these games are set in places where strange things happen, the players want in on the power of magic, too. Thus, a wide set of spells are developed for the game. These range from the widely useful, such as throwing fireballs, to spells that are only effective against blue-eyed dragons. For all of their wonderful memories and lightning-fast computational abilities, computers are incredibly stupid and teaching them how to handle all of these spells is a frightening thought indeed.

Thus, something new was needed. So, with your kind permission, I present a new set of rules to be used freely by anyone who cares to: EAMON!

Eamon is a world at the center of another galaxy—instead of revolving around any star, all of the stars revolve around it! Those huge bodies exert strange effects on the world of Eamon, bending light, gravity, time, even the laws of nature themselves! The 'normal' world of Eamon is a rough, bruising, swords-and-sorcery society where the standard job is fighting Black Knights and dragons. However, at times reality shifts and you may find yourself facing Billy the Kid or Darth Vader!

Your character has three basic numbers that determine where his or her physical abilities lie. These attributes are all rolled on 3D8, and represent your Hardiness, Agility, and Charisma.

(A special format is used to describe random numbers. It is written xDy, and means that you generate 'x' random numbers from 1 to 'y', and sum them. Thus, 3D8 gives you a number from 3 to 24. This notation comes from the strange dice used in manually-run FRP games where there are indeed eight-sided dice.)

Your Hardiness is a combination of strength and constitution. It has two major effects: it determines how much damage your body can take before it dies, and it limits the weight you can carry. A player can carry up to ten times his/her weight in gronds. A grond is a measure of weight, which is also subdivided into ten dos. With the stellar tides varying weights so much, the ex-



act metric equivalent of a grond cannot be determined; however, an average weapon weighs about three gronds, and a coin weighs about a do. If your character has the maximum Hardiness of 24, he/she can carry a great deal, which makes a do about nothing.

Your Agility measures how quick your reflexes are, as well as how accurate. High Agility helps you in combat.

Your Charisma may be the most important attribute. Although it is strongly related to physical beauty, Charisma also includes how persuasive a talker you are, and other things. High Charisma will get you lower prices on your purchases, higher prices on your sales, and help you make friends among the denizens of the dungeons.

All three of your basic attributes will not normally change through adventuring, although special magic items may raise or lower them. Although the average value of your attributes should be around 16 or 17, the normal citizen of Eamon has Hardiness, Agility, and Charisma all equal to 10. This is to reflect the fact that only people who are a cut above others would normally go adventuring.

At least in the parts of Eamon that you will be exploring, Eamon is a rough and violent world. Thus, the mechanics of combat are very important.

Roughly speaking, all weapons can be divided into five classes: axe, bow, club, spear, and sword. These all attack in different fashions, and knowing how to use one type of weapon (such as a club) does not help you use another (such as a bow). Furthermore, some weapons are easier for the inexperienced person to use than others (the novice who picks up a club is more likely to hit his target than if he uses a bow and arrow).

Thus, we have the concept of weapon expertise. For each type of weapon, you will have a number that is added to your chance of hitting. These numbers are different for each type of weapon, and start out at the following values: Axe = 5%, bow = -10%, club = 20%, spear = 10%, and sword = 0% weapon expertise.

Your weapon expertise will rise as you learn how to use your weapons. Every time you successfully strike a blow, it may be possible to learn something new about the way to use that weapon. Since obviously all you can learn about is what keeps you from being perfect, your chance of learning from a blow is the same as your chance of missing. When you learn something, your weapon expertise in the class you are using will

go up by 2%.

The second factor is the quality of the weapon, called the 'weapon complexity.' A weapon that has been forged with good balance and of fine quality steel is more likely to hit than a cheaply made rush job. (Of course, it will cost more too!) Basic weapons can be bought with either 10% complexity (good quality), 0% (fair), or -10% (poor). Weapons with higher complexities still may be found in the caves and buildings around Eamon, and you may take them as soon as you kill their previous owners.

The third major factor that affects your chance of hitting is your agility. Twice your agility is added to your chance to hit. Except under the influence of magic, under which all things are possible, this will never change.

The last normal adjustment to your chance of a successful blow is based on the armor you are wearing. Armor will usually absorb some of the damage of blows that strike your body. However, there being no free lunch, you pay for this protection by lowering your chance to hit (due to the constricting effects of the armor that you are wearing).

There are three different types of armor you can wear—Leather, which will absorb one hit from every blow but lowers your chance to hit by 10%; Chain which absorbs two hits but lowers your chance of hitting by 20%; and Plate which will take five hits in every blow but lowers your chance of hitting by a full 60%. With any of these armors (or with no armor at all), you may also carry a shield which absorbs an additional hit per blow, while lowering your chance to hit by 5%.

These adjustments are for the player who has never worn any armor before in his/her life. Experience can train a person to compensate for these effects. Thus, each player also has an armor expertise, which starts at zero, and rises by 2% jumps just as weapon expertise does. However your armor expertise can only cancel out the effects of armor—if you are wearing Leather armor and your armor expertise is 10%, any further attacks will not raise your expertise, unless you put on more armor. Furthermore, if you have built your armor expertise up to 20% while wearing chain armor and you move back to Leather, the net effect of armor will be 0, not increasing your chance to hit by 10%!

Once your total chance to hit has been determined, the computer will generate a number from 1 to 100. If the number is less than or equal to your chance to

hit, you got him! Now the question is, what did you do?

Both players and monsters (the generic term for non-player characters in Eamon—not all monsters are bad!) have a Hardiness figures, which is from 3 to 24 for humans. This number represents, among other things, the amount of damage the fellow can take before dying. On every successful blow, a random number is determined—this number varies according to the weapon and is expressed in NDN format. Any damage absorbed by armor is subtracted and if the result is greater than zero, it is taken off the target's Hardiness. This damage is not permanent, as it can be healed by either appropriate magic or rest at the player's home. However, if the player's Hardiness ever hits zero, he/she dies.

All of that, of course, is what usually happens. Sometimes, however, your weapon strikes a sensitive spot, making a critical hit. Sometimes your weapon strikes the ground, making a fumble.

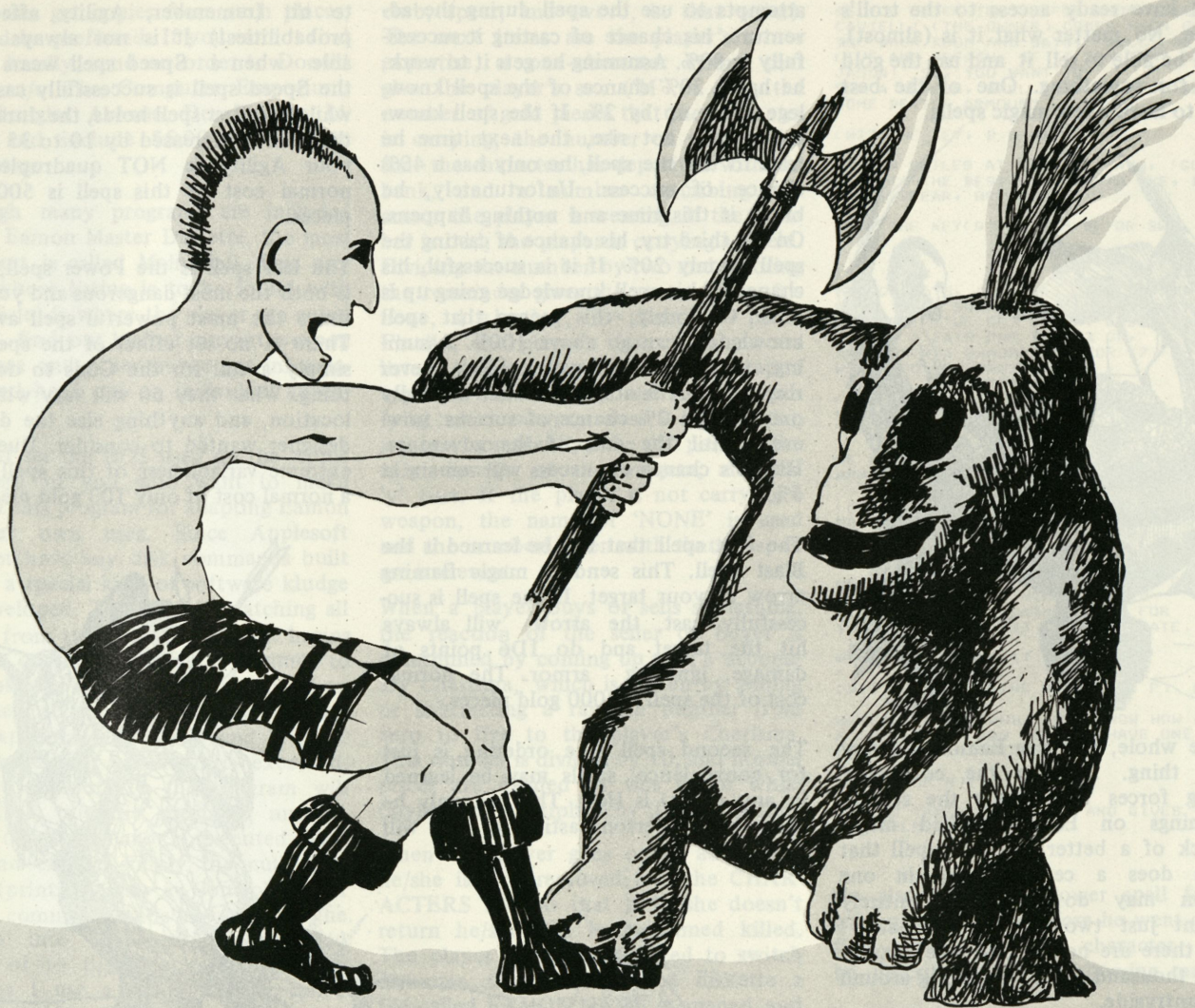
Fumbles take place about 4% of the time, regardless of your chance to hit. What happens next is up to the gods: 35% of the time the fumble is recovered from with no bad effects; 40% of the time the weapon is dropped. (Monsters are subject to fumbles. If a monster is using his natural weapons, such as teeth or claws, and drops his weapon, he simply recovers.) The weapon being used is broken 20% of the time, with a 50% chance of hurting the one using it. The last 5% of the time, the user strikes himself.

Critical hits take place 5% of the time, again no matter what the chance to hit is. A critical hit will slip through your opponent's armor 50% of the time so that he receives all of the damage done (the armor does not absorb any hits this round). Otherwise, a critical hit does more than usual damage: three-halves of normal damage is done 35% of the time, twice normal 10% of the time, and three times normal 4% of the time. One percent of all critical hits have the effect of instantly killing the opponent.

To see all of this in action, hear now the tale of Hedric and the Troll.

Hedric was a starting character with a Hardiness of 18 and an Agility of 20. (Unfortunately, he had a Charisma of 4, but that's not relevant here.) Being a beginning character, he spent his starting gold to buy a good quality axe (complexity = 10%) which does 1D6 of damage. He also bought himself a suit





of leather armor. Thus, his chance to hit was 5 (weapon expertise) + 10 (weapon complexity) + 40 (twice his Agility) - 10% (the effect of armor), or 45%.

Hedric was strolling down a dark corridor when suddenly he was attacked by a fierce Troll. Although Hedric didn't know this, the troll had a Hardiness of 30. He used his fists to do 1D4 with a 40% chance to hit. The troll's rock-hard skin acted as a natural armor, taking 2 hits per blow. (All monsters have worn their armor long enough to have raised their armor expertise to the maximum; thus, no calculations for armor are done.)

It took Hedric no time at all to realize that he had to fight this thing, and the two began attacking. In the first round, the computer came up with a 23 for Hedric's battle roll, so he hit! Unfortunately, only one point of damage was done. The troll's armor took this with ease and the troll was unhurt. However, there was a 55% chance of axe expertise and armor expertise going up by 2%.

The computer generated random numbers of 12 and 54, so both attributes did indeed go up. Thus, on his second blow, Hedric's chance of hitting will be 49%.

Simultaneously, the troll attacked Hedric. However, his battle roll was 100—a fumble! The computer gives a random number, checks its table, and comes up with the result of a dropped weapon. Since the Troll cannot drop his fists (well, you know what I mean), he recovers without any problem.

The battle continues. Hedric makes most of his blows strike home, raising his armor expertise to 10% and his axe expertise to 8%. Thus, his new chance to hit is 58%. He has gotten 20 points through the troll's armor, leaving him with 10 points.

However, the troll has not been lagging either. Hedric has taken 15 blows, leaving him with only 3 points before death! He has also managed to raise his chance of hitting to 52% (monsters also learn from experience).

None of this, of course, does Hedric know. He has been told that the troll is 'very badly injured' and that he himself is 'at death's door, knocking loudly.' He has no inkling how much, if any, his armor and weapon expertise have risen.

On the next round, Hedric gets a critical hit! The result is doubling damage. Hedric does 11 points of damage, which after 2 are taken off for armor, leaves the troll with only 1 point. The computer checks to see if his axe expertise goes up (his armor expertise cannot, since it is at its maximum), and the troll gets his attempt.

Luckily for our hero, the troll fumbles again, this time breaking his weapons. As his fists disintegrate, one of the pieces hits him in the forehead, doing two points of damage. (After all, this is FANTASY role-playing.) This brings the troll's Hardiness down to -1, and he is dead.

Now that Hedric has defeated the troll,



he will have ready access to the troll's treasure. No matter what it is (almost), he will be able to sell it and use the gold pieces for something. One of the best uses is to learn a few magic spells.



attempts to use the spell during the adventure, his chance of casting it successfully is 80%. Assuming he gets it to work, he has a 20% chance of the spell knowledge going up by 2%. If the spell knowledge does not rise, the next time he tries to use the spell, he only has a 40% chance of success. Unfortunately, he blows it this time and nothing happens. On his third try, his chance of casting the spell is only 20%. If it is successful, his chance of his spell knowledge going up is 80%. Obviously, this means that spell knowledges can go above 100%. Assuming that Hedric's spell ability never rises, the sixth attempt would normally only have a 2% chance of success; however, until the end of the adventure, Hedric's chance of success will remain at 5%.

The first spell that can be learned is the Blast spell. This sends a magic flaming arrow at your target. If the spell is successfully cast, the arrow will always hit the target and do 1D6 points of damage, ignoring armor. The normal cost of the spell is 3000 gold pieces.

The second spell (the ordering is just for convenience; spells may be learned in any order) is Heal. This can only be used on the person casting it, and will

On the whole, magic in Eamon is a very tricky thing. Due to the constantly shifting forces that power the strange happenings on Eamon (called magic for lack of a better word), a spell that always does a certain thing in one location may do something entirely different just two-hundred feet away! Thus, there are no Gandalf-like wizards with a thousand spells wandering around the countryside.

However, after years of careful experimenting, four spells have been developed that do work in almost every part of the normal Eamon world. (In the parallel worlds that one can fall into, the laws of magic are usually quite different, if magic exists at all!)

Any player who can hire a wizard to teach him/her can learn these spells. For each spell, a player gets a spell knowledge between 25% and 75%. This is his/her chance of successfully casting the spell the first time. Every time the spell is successfully cast, there is a chance of the spell knowledge going up by 2%, just as with weapon expertise. However, every time a spell is attempted (successfully cast or not!), the chance of casting that spell is cut in half for the rest of the day, down to a minimum of 5% chance of casting. There is also a 1% chance of overloading the brain and forgetting the spell completely every attempt.

For example, assume that Hedric has, through a lucky starting knowledge and some experience, brought his knowledge of a spell to 80%. The first time he

restore 1D10 points of damage. Of course, you cannot cure more points of damage than you have taken. The normal cost is 1000 gold pieces.

The third spell is Speed. This spell will double your Agility for 10 to 35 turns (a turn being the time to do any action). Its major effect is to increase your chance

to hit (remember, Agility affects hit probabilities!) It is not always noticeable when a Speed spell wears off. If the Speed spell is successfully cast again while the first spell holds, the duration of the spell is increased by 10 to 35 turns—your Agility is NOT quadrupled. The normal cost of this spell is 5000 gold pieces.

The last spell is the Power spell, and it is both the most dangerous and yet sometimes the most powerful spell available. There is no set effect of the spell; it is simply a call for the Gods to do something. What they do will vary with time, location, and anything else the dungeon designer wanted to consider. Due to the extreme variableness of this spell, it has a normal cost of only 100 gold pieces.



So far, Eamon has only been implemented on the Apple II (due to the fact that it is the computer I own). It requires either an Apple II Plus or an Apple II with the Applesoft ROM card. A disk is definitely required. Although some of the parts of Eamon can be run on a 32K system, the assumption for dungeon designers is for 48K.

One diskette called the 'Eamon Master Diskette' is needed. Additionally, each adventure scenario goes on a separate diskette. At the moment I know of five additional adventure diskettes. Finally, there is a diskette called the 'Dungeon Designer Diskette' that lets people create their own scenarios almost easily. These diskettes are being legitimately freely traded in many places, including the program libraries of computer clubs such as the Apple Pi club of Denver, Colorado.



If you can get copies from such places, it would be preferred. If you cannot find Eamon locally, send an order to Donald Brown, c/o The Computer Emporium, 3711 Douglas Avenue, Des Moines, IA 50310 and include \$5.00 for diskette and postage.

Although many programs are included on the Eamon Master Diskette, the most important is called Main Hall. This program (whose listing is to be found with this article) is where the player can equip himself, find out his/her current weapon expertises and other important information, and head out on individual adventures.

(A word on the DOS used on the Apple for those of you who want to make sense of this program for adapting Eamon to your own uses. Since Applesoft does not have any disk commands built into it, a special kind of software kludge was developed. The DOS is watching all output from the program. If a line begins with a Control-D, the line is assumed to be a disk command and is not printed on the video screen. Two commands that need explaining are READ and WRITE. When a READ command is executed, all of the inputs to the program will come from the file specified, until another disk command is executed. This command can be simply the null command (printing just a Control-D). A WRITE command sends all output of the program into the text file specified instead of to the video screen. In my programs I use a string variable called DK\$ to hold a Control-D, so that any print statements that begin with DK\$ are flagged as disk commands.)

When the MAIN HALL program is entered, there are two important data files expected on the Master Diskette. The first is called THE ADVENTURER. It is a sequential file. It simply has the name of the current player and the record number of the CHARACTERS file where his/her data are stored.

The second important file is called CHARACTERS. It is a random-access file with a length of 150 bytes per record. Record zero holds the total number of records that have been used. In each subsequent record, the first item in the record is a string. If it is the null string (" "), it is a signal to the program that this record is empty and can be used by the NEW CHARACTERS program to place a new player into.

If the first string in a record is not the null string, it is the name of the player. After that are stored the numbers that describe the player—Hardiness, Agility, Charisma, the four spell abilities (Blast, Heal, Speed and Power, in that order), and the five weapon abilities (axe, bow,

club, spear, and sword, in that order). The next items are the player's armor expertise, a one-character string that give the player's sex ('M' or 'F'), the number of gold pieces that the character is carrying, the number of gold pieces that the character has deposited with the bank, and a number describing the armor the player is wearing. If the number is odd, the player is carrying a shield. Dividing the number by two and ignoring the remainder gives the armor worn: 0 = none, 1 = leather, 2 = chain, 3 = plate. Finally, the player's four weapons are listed, giving for each the name, weapon type (a number from one to five, signifying which weapon class it belongs to), weapon complexity, the 'x' part of the xDy damage of the weapon, and the 'y' part. If the player is not carrying a weapon, the name of 'NONE' is used and the numbers given with that weapon are irrelevant.

When a player buys or sells something, the reaction of the seller or buyer is determined by coming up with a 'secondary Charisma,' which is gotten by adding or subtracting a random number from zero to five to the player's Charisma. This number is divided by 10, and normal prices are divided by this factor while buying and multiplied by it while selling.

When the player goes on an adventure, he/she is first removed from the CHARACTERS file, so that if he/she doesn't return he/she will be presumed killed. The player is then instructed to switch diskettes. On the adventure diskette a file called EAMON.NAME is opened, and the adventure name is read in (this is why there is only one adventure per disk). The computer then creates a file called FRESH MEAT into which he first writes in the record that the player used to occupy in the CHARACTERS file, then all of the information that was in that record.

After the adventure is finished, the program must have the player switch diskettes back. If the player died, all that must be done is to run 'THE WONDERFUL WORLD OF EAMON,' which will start up a new character. However, if the player survived, the program must recreate the file of THE ADVENTURER, restore the player's record in CHARACTERS, with the altered information (hopefully higher weapon expertises and more gold), and run the MAIN HALL program.

To get a feel for EAMON, observe the terrible struggles of Jason of the Boar Ring—

Jason was a starting character with a Hardiness of 12, Agility of 21, and Charisma of 14. When he got to the Main Hall, he first decided to buy a weapon.

AS YOU ENTER THE WEAPON SHOP, MARCOS CAVIELLI (THE OWNER) COMES FROM OUT OF THE BACK ROOM AND SAYS, 'WELL, AS I LIVE AND BREATHE, IF IT ISN'T MY OLD PAL JASON! SO, YOU WANT TO BUY A WEAPON, SELL A WEAPON, OR GET SOME BETTER ARMOUR?

(HIT THE KEY, B S OR A) B

MARCOS SMILES AT YOU AND SAYS, 'GOOD. I GOTTA THE BEST. YOU WANTA AXE, BOW, MACE, SPEAR, OR SWORD?

(HIT THE KEY(S), A B M SP OR SW) SWORD

MARCOS SAYS, 'WELL, I JUST HAPPEN TO HAVE THREE SWORDS IN, OF VARYING QUALITY. I'VE GOT A VERY GOOD ONE FOR 67 GP, A FAIR ONE FOR 33 GP, AND A KINDA SHABBY ONE FOR 17 WHICH DO YOU WANT?'

(HIT THE KEY, G F OR P) G  
MARCOS HANDS YOU YOUR WEAPON AND TAKES THE PRICE FROM YOU.

HE NOW ASKS YOU, 'NOW HOW ABOUT SOME ARMOUR?'

(HIT Y OR N) Y  
MARCOS TAKES YOU TO THE ARMOUR SECTION OF HIS SHOP AND SHOWS YOU SUITS OF LEATHER ARMOUR, CHAIN ARMOUR, AND PLATE.

HE SAYS, 'I CAN PUT YOU IN ANY OF THESE VERY CHEAPLY. I NEED 67 GOLD PIECES FOR THE LEATHER, 167 FOR THE CHAIN, AND 333 FOR THE PLATE.

WELL, WHAT WILL IT BE?'

(HIT N FOR NOTHING OR L C OR P) N

MARCOS SMILES AND SAYS, 'NOW HOW ABOUT A SHIELD? I CAN LET YOU HAVE ONE FOR ONLY 33 GOLD PIECES!'

(HIT Y OR N) Y

MARCOS TAKES YOUR GOLD AND GIVES YOU A SHIELD.

He also bought a Power spell for 53 gold pieces. Thus, before he went on his first adventure, his character could be described in the following manner:

YOU ARE THE MIGHTY JASON

YOUR ATTRIBUTES ARE:  
HD=12 AG=21 CH=14

YOU KNOW THE FOLLOWING SPELLS--  
POWER

YOUR WEAPON ABILITIES ARE--  
AXE BOW CLUB SPEAR SWORD  
5% -10% 20% 10% 0%

ARMOUR: SKIN AND SHIELD AE=0%

GOLD IN HAND=47, BANK=0

WEAPONS CMLPX DAM  
SWORD 10% 1 D 8  
(HIT ANY KEY TO CONTINUE)

He went to adventure in the local testing ground called 'The Beginners' Cave,' which is where almost all adventurers first go. He was in a large chamber when things started happening—

YOUR COMMAND?E  
YOU ARE IN A SMALL SIDE CHAMBER TO THE EAST OF THE LARGE CHAMBER. IT IS VERY COLD HERE, AND THE ONLY LIGHT COMES IN DIMLY FROM THE LARGE CHAMBER.  
THERE IS A GRIZZLED OLD HERMIT (WHO SMELLS AS IF HE HASN'T TAKEN A BATH IN FORTY YEARS) IN THE ROOM.  
THERE IS A BOTTLE HERE WITH A STRANGE POTION INSIDE!

YOUR COMMAND?SMILE  
NOTHING HAPPENS.  
YOU ARE STANDING IN THE EAST SIDE CHAMBER  
HERMIT IS HERE.  
YOU SEE BOTTLE



(Citizens of Eamon are rather simple-minded; if they like you they will follow you around and attack your enemies. Otherwise, they will attack you!)

YOUR COMMAND?GET BOTTLE  
GOT IT.  
YOU ARE STANDING IN  
THE EAST SIDE CHAMBER  
HERMIT IS HERE.  
YOUR COMMAND?W  
YOU ARE STANDING IN  
A LARGE CHAMBER  
HERMIT IS HERE.

YOUR COMMAND?W  
YOU ARE IN A SMALL SIDE CHAMBER TO THE  
WEST OF THE LARGE CHAMBER. IT IS VERY  
COLD HERE, AND THE ONLY LIGHT COMES IN  
DIMLY FROM THE LARGE CHAMBER.  
THERE IS A HUGE BLACK RAT WITH SHARP  
WHITE TEETH HERE!  
THERE IS A HUGE BROWN RAT WITH SHARP  
WHITE TEETH HERE!  
THERE IS A HUGE TAN RAT WITH SHARP WHITE  
TEETH HERE!  
HERMIT IS HERE.  
THERE IS A PILE OF DIAMONDS HERE!

YOUR COMMAND?ATTACK RAT  
JASON ATTACKS TAN RAT  
--A CRITICAL HIT!  
TAN RAT IS DEAD!  
BLACK RAT ATTACKS JASON  
--A MISS.  
BROWN RAT FLEES OUT AN EXIT.  
HERMIT ATTACKS BLACK RAT  
--A MISS.  
YOU ARE STANDING IN  
THE WEST SIDE CHAMBER  
BLACK RAT IS HERE.  
HERMIT IS HERE.  
YOU SEE DIAMONDS  
YOU SEE DEAD TAN RAT

YOUR COMMAND? ATTACK RAT  
JASON ATTACKS BLACK RAT  
--A MISS.  
BLACK RAT FLEES OUT AN EXIT.  
YOU ARE STANDING IN  
THE WEST SIDE CHAMBER  
HERMIT IS HERE.  
YOU SEE DIAMONDS  
YOU SEE DEAD TAN RAT

(Rats aren't too courageous, are they?)

YOUR COMMAND?GET ALL  
DIAMONDS TAKEN.  
DEAD TAN RAT TAKEN.  
YOU ARE STANDING IN  
THE WEST SIDE CHAMBER  
HERMIT IS HERE.

He and his new friend the Hermit had no trouble mopping up on the two rats that had escaped them. He also made another friend, a fighter named Heinrich. All was going very well for Jason before he entered this small room --

YOUR COMMAND?E  
YOU ARE IN A SMALL, STARK CELL WITH A  
DOOR ON THE WEST SIDE.  
HERMIT IS HERE.  
HEINRICH IS HERE.  
THERE IS A LARGE CHEST IN THE CENTER OF  
THE ROOM.

YOUR COMMAND?OPEN CHEST  
AS YOU APPROACH THE CHEST, IT SUDDENLY  
BECOMES ALIVE AND TWO TENTACLES COME OUT  
OF THE SIDE, HOLDING YOU FAST. ANOTHER  
TENTACLE RISES FROM THE TOP AND ATTEMPTS  
TO HURT YOU.  
YOU ARE STANDING IN  
AN EAST CELL  
HERMIT IS HERE.  
HEINRICH IS HERE.  
A CHEST-SHAPED MONSTER IS IN THE ROOM,  
HOLDING YOU WITH TWO TENTACLES AND  
ATTEMPTING TO DISASSEMBLE YOU WITH A  
THIRD.

(Oops!)

YOUR COMMAND?ATTACK CHEST  
ATTACK WHO?  
YOU ARE STANDING IN  
AN EAST CELL  
HERMIT IS HERE.  
HEINRICH IS HERE.  
MIMIC IS HERE.

YOUR COMMAND?ATTACK MIMIC  
JASON ATTACKS MIMIC  
--A HIT!  
MIMIC IS VERY BADLY INJURED.  
HERMIT ATTACKS MIMIC  
--A MISS.  
HEINRICH ATTACKS MIMIC  
--A MISS.  
MIMIC ATTACKS HERMIT  
--A MISS.  
YOU ARE STANDING IN  
AN EAST CELL  
HERMIT IS HERE.  
HEINRICH IS HERE.  
MIMIC IS HERE.

YOUR COMMAND?ATTACK MIMIC  
JASON ATTACKS MIMIC  
--A MISS.  
HERMIT ATTACKS MIMIC  
--A FUMBLE!  
WEAPON DROPPED!  
HEINRICH ATTACKS MIMIC  
--A MISS.  
MIMIC ATTACKS HERMIT  
--A MISS.  
YOU ARE STANDING IN  
AN EAST CELL  
HERMIT IS HERE.  
HEINRICH IS HERE.  
MIMIC IS HERE.  
YOU SEE AXE

YOUR COMMAND?ATTACK MIMIC  
JASON ATTACKS MIMIC  
--A CRITICAL HIT!  
MIMIC IS DEAD!  
AS THE MIMIC DIES, IT ROLLS OVER AND  
YOU FIND A RING UNDERNEATH IT.  
YOU ARE STANDING IN  
AN EAST CELL  
HERMIT IS HERE.  
HEINRICH IS HERE.  
YOU SEE AXE  
YOU SEE RING  
YOU SEE DEAD MIMIC

YOUR COMMAND?GET ALL  
AXE TAKEN.  
RING TAKEN.  
DEAD MIMIC IS TOO HEAVY  
YOU ARE STANDING IN  
AN EAST CELL  
HERMIT IS HERE.  
HEINRICH IS HERE.  
YOU SEE DEAD MIMIC

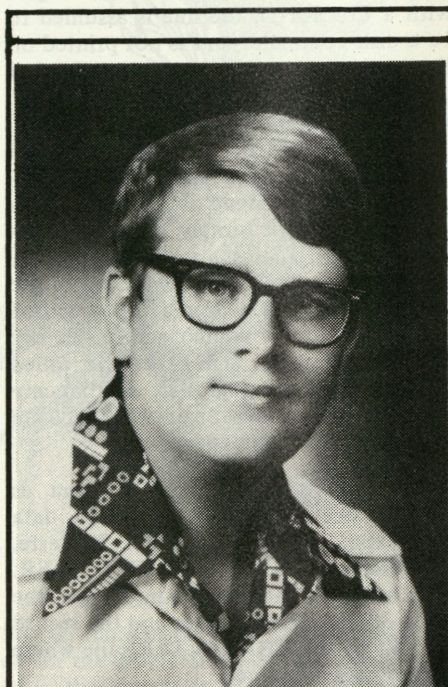
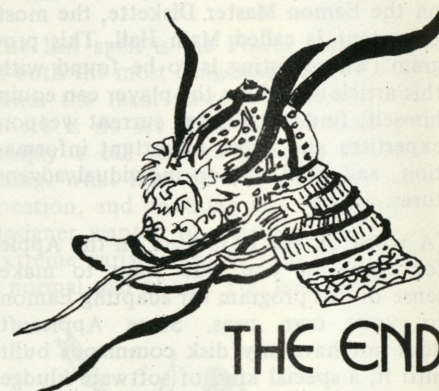
Eventually, though, Jason made it out of the adventure with his treasures intact. He sold them and had himself analyzed again. This time, his attributes and abilities were as follows:

<HIT THE KEY FOR YOUR CHOICE, 1-6>5  
YOU ARE THE MIGHTY JASON  
YOUR ATTRIBUTES ARE:  
HD=12 AG=21 CH=14  
YOU KNOW THE FOLLOWING SPELLS--  
POWER  
YOUR WEAPON ABILITIES ARE--  
AXE BOW CLUB SPEAR SWORD  
5% -10% 20% 10% 6%  
ARMOUR: SKIN AND SHIELD RE=6%  
GOLD IN HAND=657, BANK=0  
WEAPONS CMPLX DAM  
SWORD 10% 1 D 8  
AXE 0% 1 D 6

Notice that through experience both sword ability and armor expertise went up by 6%, or increased three times. Until Jason buys heavier armor, his armor expertise cannot increase any more. He also has 657 gold pieces, which should permit him to buy a new spell.

Although Eamon has a reasonably complete background set, there is still room for infinite expansion. Jason may have been able to handle small problems like the rats and survive little traps like the chest-mimic, but surely in the depths of your imagination you have nastier things in mind! Furthermore, so far all of the adventures written for Eamon

have resembled the Woods and Crowther's ADVENTURE, but Eamon is more flexible than that. Consider a game where you are leading an army into battle, with morale affected by your Charisma! The possibilities are endless. There is a full universe out there, and it is up to you to pull it into your computer!



Donald Brown became a personal computing fanatic almost three years ago when his father brought home one of the first Apple IIs. He also permitted the infamous William Fesselmeyer to lure him into fantasy games over two years ago. He now spends his time in Des Moines writing computer programs, playing games, and incidentally working towards a Bachelor's Degree at Drake University. He can best be contacted through the Computer Emporium of Des Moines.



# Listings

```

0 REM THE WONDERFUL WORLD OF
1 REM EMMON
2 REM
3 REM MAIN HALL (BUY SPELLS
4 REM AND EQUIPMENT AND
5 REM GO ON ADVENTURES)
6 REM
7 REM BY DONALD BROWN
8 REM
9 REM
10 DK$ = CHR$(4)
20 ONERR GOTO 7000
30 PRINT DK$;"OPEN THE ADVENTURE
R,SG,D1"; PRINT DK$;"READ TH
E ADVENTURER"; INPUT NAMES; INPUT
REC; PRINT DK$;"CLOSE"
40 ONERR GOTO 100
50 PRINT DK$;"OPEN CHARACTERS,L1
50"; PRINT DK$;"READ CHARACT
ERS,R"; REC; INPUT NAME$; INPUT
HD,AG,CH; FOR X = 1 TO 4: INPUT
SA$(X); NEXT X; FOR X = 1 TO
5: INPUT WA$(X); NEXT X; INPUT
RE
60 INPUT SEX$
70 INPUT GOLD,BANK; INPUT AC; FOR
W = 1 TO 4: INPUT NAME$(W);
WTYPE$(W); WOODS$(W); WIDEX$(W
); WSIDES$(W); NEXT W; PRINT
DK$;"CLOSE"
80 DEF FN P(X) = INT (X * RTIO
+ 5)
90 PRINT : PRINT "AS YOU WANDER
ABOUT THE HALL, YOU"; PRINT
"REALIZE YOU CAN DO ONE OF SE
IX THINGS--"; PRINT "1. G
O ON AN ADVENTURE."; PRINT "
2. VISIT THE WEAPON SHOP
FOR WEAPONS
AND/OR ARM
OUR."
110 PRINT "3. HIRE A WIZARD T
O TEACH YOU SOME
SPE
LLS."; PRINT "4. FIND THE
"
BANKER TO DEPOSIT OR"; PRINT
WITHDRAW SOME GOLD.";
PRINT "5. EXAMINE YOUR A
BILITIES."; PRINT "6. TEM
PORARILY LEAVE THE UNIVERSE.
"
120 PRINT : PRINT "<HIT THE KEY
FOR YOUR CHOICE, 1-6>";
130 GET A$; CMD = VAL (A$); IF C
MD < 1 OR CMD > 6 THEN 130
140 PRINT CMD; PRINT : ON CMD GOTO
1000,2000,3000,4000,5000,600
0
200 PRINT NAME$; PRINT HD; PRINT
AG; PRINT CH; FOR X = 1 TO 4
: PRINT SA$(X); NEXT X; FOR X
= 1 TO 5: PRINT WA$(X); NEXT
: PRINT RE; PRINT SEX$
: PRINT GOLD; PRINT BANK; PRINT
AC; FOR W = 1 TO 4: PRINT WN
AME$(W); PRINT WTYPE$(W); PRINT
WOODS$(W); PRINT WIDEX$(W); PRINT
WSIDES$(W); NEXT W
2650
2670 PRINT : PRINT "MARCOS SAYS,
'WELL, I JUST HAPPEN TO'; PRINT
'HAVE THREE '.N$;'S IN. OF V
ARYING"; PRINT "QUALITY. I'V
E GOT A VERY GOOD ONE FOR"; PRINT
FN P(2 * BP); "GP, A FAIR O
NE FOR"; FN P(BP); "GP,"
2680 PRINT "AND A KINDA SHABBY O
NE FOR"; FN P(BP / 2); PRINT
"WHICH DO YOU WANT?"; PRINT
: PRINT "<HIT THE KEY, G F O
R P>";
2690 GET A$; FOR Q = 1 TO 3: IF
A$ < > MID$( "GFP",0,1) THEN
NEXT; GOTO 2690
2100 PRINT A$; P = BP + BP * (Q =
1) - BP / 2 * (Q = 3); P = FN
P(P)
2110 IF P > GOLD THEN PRINT : PRINT
"MARCOS SHAKES A FINGER AT Y
OU AND SAYS."; PRINT "YOU S
HOULDN'T PLAY TRICKS ON AN O
LD"; PRINT "FRIEND! COME BAC
K WHEN YOU GOTTA MORE"; PRINT
"GOLD OR YOU WANT SOMETHING
YOU CAN"; PRINT "AFFORD. HE
THEN SHOOS YOU OUT THE"; PRINT
"DOOR."; PRINT : GOTO 100
2120 FOR W = 1 TO 4: IF UN$(W) =
N$ THEN N$ = N$ + "#"; GOTO
2120
2130 NEXT W; FOR W = 1 TO 4: IF
UN$(W) < > "NONE" THEN NEXT
UN$(W) = N$; WT$(W) = TP; WOX(
W) = 10 * (2 - Q); WDX$(W) = D
ICE; WIS$(W) = SIDES; GOLD = GO
LD - P
2150 PRINT "MARCOS HANDS YOU YOU
R WEAPON AND TAKES"; PRINT "
THE PRICE FROM YOU."; PRINT
2160 IF GOLD < FN P(50) THEN 22
10
2170 PRINT "HE NOW ASKS YOU, 'NO
W HOW ABOUT SOME'; PRINT "AR
MOUR?"; PRINT : PRINT "<HIT
Y OR N>";
2180 GET A$; IF A$ = "N" THEN PRINT
A$; PRINT : GOTO 2210
2190 IF A$ < > "Y" THEN 2180
2200 PRINT A$; GOTO 2490
2210 PRINT "MARCOS SMILES AND SA
YS, 'COME BACK'; PRINT "AGAI
N SOON! AS HE SHOOS YOU OUT
OF HIS"; PRINT "SHOP."; PRINT
: GOTO 100
2220 PRINT : PRINT "MARCOS ASKS
YOU, 'IS THIS WEAPON YOU'; PRINT
'WANT TO SELL ME A STANDARD
WEAPON LIKE'; PRINT "I SELL?
(CHIT 'Y' OR 'N')";
2230 GET A$; IF A$ = "Y" THEN 23
50
2240 IF A$ < > "N" THEN 2230

```



```

2250 PRINT A$: PRINT P = INT (
50 / RTIO + .5); PRINT "MARCO
OS SAYS, 'THEN I CAN ONLY GI
VE YOU'"; PRINT P; "GOLD PIEC
ES FOR IT, TAKE IT OR"; PRINT
"LEAVE IT!" ; PRINT : PRINT
" (HIT T OR L) " ;
2260 GET A$: IF A$ = "L" THEN PRINT
A$: PRINT : GOTO 2210
2270 IF A$ < > "T" THEN 2260
2280 PRINT A$: PRINT : PRINT "MA
RCOS SAYS, 'OKAY, WHAT'VE YO
U GOT?"; PRINT : INPUT "CENT
ER THE WEAPON NAME"; W$: FOR
W = 1 TO 4: IF W$ (W) < > W
$ THEN NEXT : PRINT : PRINT
"MARCO FROWNS AT YOU AND SA
YS, 'YOU'"; PRINT "CAN'T SELL
A WEAPON YOU DON'T OWN!" ; PRINT
: GOTO 2210
2290 L = LEFT$ (W$, 5) = "SWORD" OR
LEFT$ (W$, 5) = "SPEAR" OR LEFT$
(W$, 4) = "MACE" OR LEFT$ (W$, 3
$) = "BOW" OR LEFT$ (W$, 3
) = "AXE"; IF L THEN PRINT
"MARCO FROWNS AT YOU AND SA
YS, 'THIS IS'; PRINT "A WEAP
ON LIKE I SELL, REMEMBER!" ;
GOTO 2420
2300 PRINT "MARCO GIVES YOU YOU
R MONEY AND TAKES"; PRINT "Y
OUR WEAPON." ; PRINT : GOLD =
GOLD + P; FOR W2 = 1 TO 4: W
$(W2) = W$ (W2 + 1); W2$ (W2) =
W2$ (W2 + 1); W2$ (W2) = W2$ (W2
+ 1); W2$ (W2) = W2$ (W2 + 1);
W2$ (W2) = W2$ (W2 + 1); NEXT
: W$ (4) = "NONE"
2310 PRINT : PRINT "MARCO ASKS
YOU, 'HOW ABOUT BUYING A'; PRINT
"WEAPON?"; PRINT : PRINT " (
HIT Y OR N) " ;
2320 GET A$: IF A$ = "Y" THEN PRINT
A$: PRINT : GOTO 2050
2330 IF A$ < > "N" THEN 2320
2340 PRINT A$: PRINT : GOTO 2160
2350 PRINT : PRINT "MARCO ASKS,
'WELL, WHAT WEAPON YOU'; PRINT
"WANTA RETURN?"; PRINT : GOSUB
2650: PRINT
2360 W2 = W: FOR W = 1 TO 4: IF LEFT$
(W$ (W), LEN (W$)) < > N$ THEN
NEXT : PRINT "MARCO LAUGHS
AND SAYS, 'YOU NEVER'; PRINT
"BOUGHT ONE FROM ME, REMEMBE
R?"; PRINT : GOTO 2210
2370 BP = BP + BP * (W2$ (T) > 5) -
BP / 2 * (W2$ (T) < - 5); P =
INT (BP / RTIO + .5); IF P >
BP / 4 THEN P = INT (BP / 4
+ .5)
2380 PRINT "MARCO EXAMINES YOUR
WEAPONS AND SAYS:"; PRINT "
'WELL, YOU'VE BANGED IT UP A
BIT, BUT"; PRINT "I CAN GIV
E YOU 'P.'; "GOLD PIECES"; PRINT
"FOR IT, TAKE IT OR LEAVE IT
.'"; PRINT : PRINT " (HIT T O
R L) " ;
2390 GET A$: IF A$ = "L" THEN PRINT
A$: PRINT : GOTO 2210
2400 IF A$ < > "T" THEN 2390
2410 PRINT A$: PRINT : GOTO 2300
2420 MID$ ("ABMS", W, 1) = LEFT$
(W$, 1) THEN ON W GOTO 2440,
2450, 2460, 2470
2430 NEXT W
2440 BP = 25: GOTO 2370
2450 BP = 40: GOTO 2370
2460 BP = 20: GOTO 2370
2470 IF MID$ (W$, 2, 1) = "P" THEN
BP = 25: GOTO 2370
2480 BP = 50: GOTO 2370
2490 A2 = INT (AC / 2); SH = AC -
A2 * 2; B1 = 25 * (A2 = 1) +
60 * (A2 = 2) + 100 * (A2 =
3); T1 = INT (B1 / RTIO + .5
); PRINT "MARCO TAKES YOU T
O THE ARMOUR SECTION"; PRINT
"OF HIS SHOP AND SHOWS YOU S
UITS OF"; PRINT "LEATHER ARM
OUR, CHAIN ARMOUR, AND PLATE
."; IF T1 > B1 * 2 THEN T1 =
B1 * 2
2500 PRINT : PRINT "HE SAYS, 'I
CAN PUT YOU IN ANY OF THESE'
"; PRINT "VERY CHEAPLY, I NE
ED"; FN P (100); "GOLD"; PRINT
"PIECES FOR THE LEATHER, "; FN
P (250); "FOR"; PRINT "THE CH
AIN, AND "; FN P (500); "FOR
THE PLATE." ; PRINT
2510 IF T1 THEN PRINT "ALSO, I
CAN GIVE YOU A TRADE-IN ON Y
OUR"; PRINT "OLD ARMOUR OF
"; FN P (100); "GOLD"; PRINT
"; T1; "GOLD PIECES." ; PRINT
2520 PRINT "WELL, WHAT WILL IT B
E?"; PRINT : PRINT " (HIT N
FOR NOTHING OR L C OR P) " ;
2530 GET A$: FOR A = 1 TO 4: IF
A$ < > MID$ ("NLOP", A, 1) THEN
NEXT : GOTO 2530
2540 PRINT A$: PRINT : IF A = 1 THEN
2570
2550 P = FN P (100 * (A = 2) + 25
0 * (A = 3) + 500 * (A = 4))
- T1; IF P > GOLD THEN PRINT
"MARCO FROWNS WHEN HE SEES
THAT YOU DO"; PRINT "NOT HAV
E ENOUGH TO PAY FOR YOUR"; PRINT
"ARMOUR AND SAYS, 'I DON'T G
IVE CREDIT!"; PRINT : GOTO
2570
2560 PRINT "MARCO TAKES YOUR OL
D ARMOUR AND YOUR"; PRINT "G
OLD AND HELPS YOU INTO YOUR
NEW"; PRINT "ARMOUR." ; PRINT
: GOLD = GOLD - P; A2 = A - 1
2570 IF SH THEN 2630
2580 PRINT "MARCO SMILES AND SA
YS, 'NOW HOW ABOUT'; PRINT "A
SHIELD? I CAN LET YOU HAV
E ONE FOR"; PRINT "ONLY "; FN
P (50); "GOLD PIECES!"; PRINT
: PRINT " (HIT Y OR N) " ;
2590 GET A$: IF A$ = "N" THEN PRINT
A$: PRINT : GOTO 2630
2600 IF A$ < > "Y" THEN 2590
2610 PRINT A$: PRINT "WHEN
FN P (50) THEN PRINT "WHEN
HE SEES THAT YOU DO NOT HAVE
"; PRINT "ENOUGH GOLD TO BUY
THE SHIELD, MARCO"; PRINT
" FROWNS AND SAYS, 'I'M SORRY
, BUT I'; PRINT "DON'T GIVE
CREDIT!"; PRINT : GOTO 2630
2620 GOLD = GOLD - FN P (50); SH =
1; PRINT "MARCO TAKES YOUR
GOLD AND GIVES YOU"; PRINT "
A SHIELD." ; PRINT : GOTO
2630
2630 AC = A2 * 2 + SH; PRINT : GOTO
100
2640 END
2650 PRINT " (HIT THE KEY'S), A B
M SP OR SW) " ;
2660 GET A$: FOR W = 1 TO 4: IF
A$ < > MID$ ("ABMS", W, 1) THEN
NEXT : GOTO 2660
2670 ON W GOTO 2680, 2690, 2700, 27
10
2680 PRINT "AXE"; BP = 25; N$ = "A
XE"; TP = 1; SIDE = 6: GOTO 27
50
2690 PRINT "BOW"; BP = 40; N$ = "B
OW"; TP = 2; SIDE = 6: GOTO 27
50
2700 PRINT "MACE"; BP = 20; N$ = "
MACE"; TP = 3; SIDE = 4: GOTO
2750
2710 PRINT A$:
2720 GET A$: IF A$ = "P" THEN PRINT
"PEAR"; BP = 25; TP = 4; SIDE =
5; N$ = "SPEAR"; GOTO 2750
2730 IF A$ < > "W" THEN 2720
2740 PRINT "WORD"; BP = 50; N$ = "
SWORD"; TP = 5; SIDE = 8
2750 DICE = 1: RETURN
2760 END
2770 PRINT : PRINT "AFTER A FEW
MINUTES DILIGENT SEARCHING, "
: PRINT "YOU FIND HOKAS TOKA
S, THE OLD MAGE, HE"; PRINT
"LOOKS AT YOU AND SAYS, 'SO
YOU WANT OLD'; PRINT "TOKEY
TO TEACH YOU SOME MAGIC, HEH
HEH?";
3010 PRINT "WELL, IT'LL COST YOU
TODAY MY FEES ARE:"; C2 = C
H + INT (11 * RND (1) - 5)
: C2 = C2 + (C2 = 0); RTIO = 1
0 / C2
3020 PRINT " BLAST "; FN P (30
00); "GP"; PRINT " HEAL "
: FN P (1000); "GP"; PRINT "
" SPEED "; FN P (5000); "GP"; PRINT
" POWER "; FN P (100); "GP"
: PRINT
3030 PRINT "WELL, WHICH WILL IT
BE?"; PRINT : PRINT " (HIT T
HE KEY FOR YOUR SPELL, B, H, S
OR P) " ;

```



```

3040 GET A$: FOR SP = 1 TO 4: IF
A$ < > MID$( "BHPSP",SP,1) THEN
NEXT : GOTO 3040
3050 ON SP GOTO 3060,3070,3080,3
090
3060 PRINT "BLAST": P = FN P(300
0): GOTO 3100
3070 PRINT "HEAL": P = FN P(1000
): GOTO 3100
3080 PRINT "SPEED": P = FN P(500
0): GOTO 3100
3090 PRINT "POWER": P = FN P(100
)
3100 PRINT : IF P > GOLD THEN PRINT
"WHEN HOKAS SEES THAT YOU DO
N'T HAVE", PRINT "ENOUGH TO
PAY HIM, HE STALKS TO THE": PRINT
"BAR, MUTTERING ABOUT YOUNGS
TERS WHO": PRINT "SHOULD BE
TURNED INTO FROGS.": PRINT :
GOTO 100
3110 IF SAK(SP) THEN PRINT "HOK
AS SAYS, 'I OUGHT TO TAKE YO
UR": PRINT "GOLD ANYWAY, BUT
HAVEN'T YOU FORGOTTEN": PRINT
"SOMETHING? I ALREADY TAUGH
T YOU THAT": PRINT "SPELL!:"
: PRINT : PRINT "SHAKING HIS
HEAD SADLY, HE RETURNS TO":
PRINT "BAR.": PRINT : GOTO
100
3120 GOLD = GOLD - P: SAK(SP) = INT
(50 * RAND(1) + 26): PRINT
"HOKAS TEACHES YOU YOUR SP
EL, TAKES": PRINT "HIS FEE, A
ND RETURNS TO HIS STOOL ON":
PRINT "THE BAR. AS YOU WAL
K AWAY YOU HEAR"
3130 PRINT "HIM ORDER A DOUBLE D
RAGON BLOMB.": PRINT : GOTO
100
3140 END
4000 PRINT : PRINT "YOU HAVE NO
TROUBLE SPOTTING SHYLOCK": PRINT
"MCFENNEY, THE LOCAL BANKER,
DUE TO HIS": PRINT "LARGE B
ELLY.": PRINT : PRINT "YOU A
TTRACT HIS ATTENTION, AND HE
COMES": PRINT "OVER TO YOU
AND SAYS, 'WELL,"
4010 PRINT NAME$: "MY DEAR ": MID$(
"GIRLBOY", (SEX$ = "M") * 4 +
1,4): "": PRINT "WHAT A PLEA
SURE TO SEE YOU! DO YOU WANT
"
4020 PRINT "TO MAKE A DEPOSIT OR
A WITHDRAWL?": PRINT : PRINT
"CHIT 'D' FOR DEPOSIT, 'W' F
OR WITHDRAWL) "
4030 GET A$: IF A$ = "D" THEN PRINT
A$: PRINT : GOTO 4090
4040 IF A$ < > "W" THEN 4030
4050 PRINT A$: PRINT : PRINT "SH
YLOCK SAYS, 'WELL, YOU HAVE
".BANK: PRINT "GOLD PIECES S
TORED WITH ME. HOW MANY": PRINT
"DO YOU WANT TO TAKE BACK?:"
: PRINT
4060 PRINT "CENTER THE NUMBER OF
GOLD PIECES TO": INPUT "
WITHDRAW) ": A$: MNY = VAL (
A$): IF MNY < 0 OR MNY <
INT (MNY) OR STR$(MNY) <
> A$ THEN PRINT : PRINT "T
HE BANKER SCOWLS AND SAYS, 'C
OME, COME.": PRINT "YOU'RE N
OT MAKING SENSE! TRY AGAIN
": PRINT : GOTO 4060
4070 IF MNY > BANK THEN PRINT :
PRINT "THE BANKER THROWS YO
U A TERRIBLE GLANCE": PRINT
"AND SAYS, 'THAT'S MORE THAN
YOU'VE GOT!": PRINT "YOU KN
OW I DON'T MAKE LOANS TO YOU
R": PRINT "KIND!" WITH THAT
HE LOSES HIMSELF IN": PRINT
"THE CROWD.": PRINT : GOTO 1
00
4080 GOLD = GOLD + MNY: BANK = BAN
K - MNY: PRINT : PRINT "THE
BANKER HANDS YOU YOUR GOLD A
ND": PRINT "SAYS, 'THAT LEAV
ES YOU WITH ":BANK: PRINT "P
IECES IN MY CARE. HE SHAKE
S YOUR": PRINT "HAND AND WAL
KS AWAY.": PRINT : GOTO 100
4090 PRINT "SHYLOCK GETS A WIDE
GRIN ON HIS FACE": PRINT "AN
D SAYS, 'GOOD FOR YOU! HOW
MUCH DO": PRINT "YOU WANT TO
DEPOSIT?:"
4100 PRINT : PRINT "CENTER THE A
MOUNT YOU WANT TO DEPOSIT":
INPUT " ": A$: MNY = VAL
(A$): IF MNY < 0 OR MNY <
INT (MNY) OR STR$(MNY) <
> A$ THEN PRINT : PRINT "T
HE BANKER SCOWLS AT YOU AND
SAYS.": PRINT "COME, COME,
YOU'RE NOT MAKING SENSE!": PRINT
"TRY AGAIN.": PRINT : GOTO
4100
4110 IF MNY > GOLD THEN PRINT :
PRINT "THE BANKER WAS VERY
PLEASED WHEN YOU": PRINT "TO
LD HIM THE SUM, BUT WHEN HE
DISCOVERED THAT YOU DIDN'T HA
VE THAT MUCH ON YOU,": PRINT
"HE WALKED AWAY, SHOUTING AB
OUT FOOLS": PRINT "WHO PLAY
TRICKS ON A KINDLY BANKER.":
PRINT : GOTO 100
4120 GOLD = GOLD - MNY: BANK = BAN
K + MNY: PRINT "SHYLOCK TAKE
S YOUR MONEY, PUTS IT IN": PRINT
"HIS BAG, LISTENS TO IT JING
LE, THEN": PRINT "THANKS YOU
AND WALKS AWAY.": PRINT : GOTO
100
4130 END

```

The End



# Enchanted House Listing Continued from pg. 8

```

2629 M=23 => EGG
2630 PRINT "A golden EGG is on the floor." : RETURN
2638
2639 M=24 => JEWEL
2640 PRINT "On the floor is a JEWEL." : RETURN
2648
2649 M=25 => URSE
2650 PRINT "A magnificent URSE is here." : RETURN
2999
3500 ----- responding to unlisted words ----
3501
3502 You could set S=0 here & switch S to one if the message
3503 were understood. You'd have 3995 ON S+1 GOTO 365. 305
3504
3505 IF A$="EXTEND TIME" THEN T9 = T9+24 : N=28 : GOTO 3315
3994
3995 GOTO 365
3999
4000 ----- details of settings & droppings ----
4002
4005 Assign M an appropriate value. (M is the # of the
4010 manifestation named by N$ (unless M = 0)).
4015 IF U=0 OR N=0 THEN M=0 : RETURN
4020 IF N=24 THEN IF L(11)=9 THEN M=11 : RETURN ELSE M=14 : RETURN
4025 FOR I=1 TO M9 : IF N1(I)=N THEN M=I : RETURN ELSE NEXT I
4030 M=0 : IF N<22 AND N<12 THEN RETURN
4035 FOR J=3 TO 5 : IF L(J)<0 THEN M=J : NEXT J ELSE NEXT J
4040 RETURN
4044
4045 IF L(M)=R AND C<09 THEN 4075 ELSE IF C=09 THEN 4055
4050 PRINT "You've still got the " ; N$ : GOTO 4140
4054
4055 PRINT "You can only carry " ; C9 ; " objects (not counting)
4060 PRINT "the WATCH) and you are already carrying that much."
4065 PRINT "If you really want to GET something else, then you"
4070 PRINT "will have to DROP something first." : GOTO 4140
4074
4075 IF W(M)=0 THEN 4090 ELSE PRINT "Congratulations, " ; P$(P)
4080 PRINT "You'll earn points when the " ; N$ ; " is DEPOSITED (or"
4085 PRINT "DROPPED) in the Bank." : C=C+1 : L(M)=1 : RETURN
4089
4090 PRINT "OK; you are now carrying the " ; N$ ; "."
4095 L(M) = -1 : C=C+1 : RETURN
4099
4100 IF L(M) = -1 THEN PRINT "OK." : L(M)=R : C=C-1 ELSE 4135
4105 IF R=0 THEN GOSUB 5175
4110 IF R<3 AND R<9 THEN RETURN
4115 IF L(2)=3 THEN L(2)=0 : L(3)=3 : L(7)=0 : L(8)=3
4120 IF L(4)=9 THEN L(4)=0 : L(3)=9 : L(11)=0 : L(13)=9
4125 IF L(5)=9 THEN L(5)=0 : L(3)=9 : L(11)=0 : L(12)=9
4130 RETURN
4134
4135 PRINT "You don't have the " ; N$ ; "."
4139
4140 PRINT "To see an inventory of what you're carrying, type"
4145 PRINT "INVENTORY." : RETURN
4299
4300 ----- details of UP and DOWN moving ----
4301
4302 This block is not used in the Nellian is Thirsty
4303 version of Enchanted House. For use, see the
4304 July-Aug. 1980 issue of RECREATIONAL COMPUTING.
4304
4305 --- setting H = (# for floor that we're on)
4310 I2=R7+R8 : FOR H=1 TO R6 : (top floor does with H=1)
4315 IF H<I2-1<R AND R<H+12 THEN 4325 ELSE NEXT H
4320 --- checking for special conditions common to U & D
4325 GOTO 4390
4390 I=1+INT((R-1)/8) : J=R-R*(I-1)
4395 IF (going UP) then 4405 else 4455
4400 --- UPing
4405 IF H=1 THEN 4445
4415 code such as ON I followed by ON J
4416 similar to the code found in lines 265- 290.
4435 PRINT "The CHAIR is too heavy to carry UP." : GOTO 305
4440 R = R-R7+R8 : GOTO 605
4445 PRINT "There's no way to GO UP here." : GOTO 305
4450 --- DOWNing
4455 This part follows the same pattern as indicated in
4457 the UPing part above.
4499
4500 ----- drawing map ----
4504
4505 I1=60 : PRINT : PRINT "You are in the " ; W4$(R) ; " Room now." ;
4510 PRINT TAB(11) ; "N" : PRINT TAB(11-2) ; "W + E"
4515 PRINT TAB(11) ; "S"
4519
4520 I2=R7+R8 : FOR I1=1 TO R6
4521 IF the MAP is DROPPed and CONSULTed, every floor is drawn.
4523 IF L(19)=1 THEN 4525 ELSE GOSUB 4530 : NEXT I1 : RETURN
4525 IF (I1-1)*I2 < R AND R <= I1*I2 THEN 4530 ELSE NEXT I1
4529
4530 K0 = (I1-1)*I2 : K9 = I1*I2 : K=K0 : J=1
4534
4535 IF K=K0 AND J=1 THEN PRINT W3$(7) ;
4540 IF K=K0 AND J=1 AND J<R8+1 THEN PRINT W3$(8) ;
4545 IF K=K0 AND J=R8+1 THEN PRINT W3$(9)
4549
4550 IF K=K0 AND K<K9 AND J=1 THEN PRINT W3$(4) ;
4555 IF K=K0 AND K<K9 AND J=1 AND J<R8+1 THEN PRINT W3$(5) ;
4560 IF K=K0 AND K<K9 AND J=R8+1 THEN PRINT W3$(6)
4564
4565 IF K=K9 AND J=1 THEN PRINT W3$(1) ;
4570 IF K=K9 AND J=1 AND J<R8+1 THEN PRINT W3$(2) ;
4575 IF K=K9 AND J=R8+1 THEN PRINT W3$(3)
4579
4580 IF J>R8 THEN 4595
4585 IF W1(K+J)>0 THEN PRINT W1$(K+J) ; ELSE PRINT W1$(0) ;
4590 J=J+1 : GOTO 4535
4594
4595 IF K=K9 THEN RETURN ELSE J=1
4599
4600 IF W2(K+J)>0 THEN PRINT W2$(W2(K+J)) ; ELSE PRINT W2$(0) ;
4605 IF J>R8 THEN PRINT : GOTO 4620
4609
4610 IF W4(K+J)>0 THEN PRINT W4$(K+J) ; ELSE PRINT W4$(0) ;
4615 J=J+1 : GOTO 4600
4619
4620 K=K+R8 : J=1 : GOTO 4535
4999
5000 ----- details of otherwise responding ----
5001
5005 for N$ in ( N, S, E, W )
5010 (barrier with R=1)
5015 IF R=1 THEN IF L(10)<-1 THEN 5020 ELSE 5030 ELSE 5040
5020 PRINT "The door is locked and you're not carrying the key."
5025 W1(4)=3.5 : GOTO 305
5030 W1(4) = 2.5 : R = 4 : GOTO 605
5035 (barrier with R=2)
5040 IF R=2 THEN IF L(6)=2 THEN 5045 ELSE 5060 ELSE 5070
5045 PRINT "You cannot enter--the doorway is blocked by a thick"
5050 PRINT "growth of carrots."
5055 W2(3)=3.5 : GOTO 305
5060 R=3 : W2(3)=2.5 : GOTO 605
5065 (barrier with R=4)
5070 IF R=4 THEN IF L(10)<-1 THEN 5020 ELSE 5080
5075 R=2 : GOTO 605
5080 W1(4)=2.5 : R=1 : GOTO 605
5085
5100 for INVENTORY
5105 PRINT "Your WATCH stays on your wrist for the whole visit."
5110 IF C=0 THEN PRINT "You aren't carrying anything else." : RETURN
5115 PRINT "Here is an inventory of what you're carrying:"
5120 FOR J=1 TO M9
5125 IF L(J)=1 THEN PRINT TAB(20) ; N$(N1(J)) ; NEXT J ELSE NEXT J
5130 RETURN
5149
5150 for SCORE
5155 GOSUB 5175 : PRINT "You have " ; I ; " points out of 100."
5160 PRINT "You earn points by DEPOSITING valuable things in " ;
5165 PRINT "the Bank." : RETURN
5170
5175 I=0 : FOR J=1 TO M9 : IF L(J)=R0 THEN I=I+W(J)
5180 IF P>1 THEN A$="all" ELSE A$=""
5185 NEXT J : IF I<100 THEN RETURN ELSE PRINT : PRINT "Mr. " ; N$(30) ;
5190 PRINT " shouts, << You " ; A$ ; "won! >>." : CHR$(7) : GOTO 620
5199
5200 for KICK
5205 PRINT "Kicking " ; N$ ; " will not help anything." : GOTO 305
5249
5250 for CONSULT
5255 IF NOT ( N=11 OR N=23 OR N=24 ) THEN 5305
5260 IF L(6)=0 THEN 5275
5265 PRINT "The " ; N$(N1(M)) ; " laughs and says, <<Well, I know"
5270 PRINT "that you know a carrot-liking RABBIT>>." : GOTO 305
5275 ON 2-W4(1) GOTO 5280, 5290
5280 PRINT "The " ; N$(N1(M)) ; " says, << " ; P$(P) ; "
5285 PRINT "I think that you're doing great >>." : GOTO 305
5290 PRINT "The " ; N$(N1(M)) ; " smiles mysteriously and says,"
5295 PRINT "<< I recommend that you SIT in the CHAIR >>."
5300 GOTO 305
5305 IF N=15 THEN GOSUB 4505 : GOTO 305
5310 IF N=20 THEN PRINT "It doesn't help here." : GOTO 305
5315 IF N=28 THEN PRINT " scoots have passed"
5320 IF N=28 THEN PRINT "& " ; T9-T ; " scoots remain." : GOTO 305
5325 GOTO 5980
5349
5350 for OPEN and UNLOCK
5355 IF N=9 THEN 5015 ELSE 5980
5399
5400 for FILL (& GET MILK)
5405 IF N<12 THEN IF N<24 THEN 5980 ELSE U=5 : N=12 : GOTO 435
5410 IF R<2 OR L(3)<-1 THEN 5425 ELSE L(3)=0 : L(4)=1
5415 PRINT "Glugs; glugs; plink. Your BOWL has cold MILK now."
5420 GOTO 305
5425 IF N=22 AND M=3 THEN PRINT "The BOWL is empty." ; N$="BOWL"
5430 IF U<13 THEN IF L(M)=R THEN GOSUB 4045 : GOTO 305
5435 IF M=4 OR M=5 THEN PRINT "The BOWL is still full of MILK."
5440 IF R<2 THEN PRINT "The Milk Room has lots of MILK."
5445 IF L(M)=1 OR M<3 OR M>5 THEN PRINT "You need a BOWL."
5450 GOTO 305
5599
5600 for LIST
5605 IF N<25 THEN 5980 ELSE PRINT "I know these verbs: " ;
5610 FOR I=1 TO U9 : A$=U$(I) : GOSUB 5655 : NEXT I
5615 PRINT "I know these nouns: " ;
5620 FOR I=1 TO N9 : A$=N$(I) : GOSUB 5655 : NEXT I
5625 PRINT "I know some one word messages (for example: HELP)"
5630 PRINT "and I know some two word messages (for example:)"
5635 PRINT "FILL BOWL, OFFER MILK). Whatever you say, I'll look"
5640 PRINT "at the first and last words; you could say"
5645 PRINT "SIT in the CHAIR or CONSULT THE SLEEPY MR. " ; N$(30) ;
5650 PRINT " " : GOTO 305
5655 PRINT A$ ;
5660 IF A$=U$(U9) OR A$=N$(N9) THEN PRINT " " ELSE PRINT " " ;
5665 IF POS(1)<55 THEN RETURN ELSE PRINT : RETURN
5699
5700 for WORTH
5705 IF M=0 THEN 5980 ELSE PRINT "The " ; N$(M) ; " is worth " ; W(M) ;
5710 PRINT " points." : IF W(M)=0 THEN PRINT "The " ; N$(M) ; " ELSE 305
5715 PRINT " might be useful as a tool." : PRINT
5720 IF N=11 OR N=23 OR N=24 THEN PRINT "Friends are valuable."
5725 IF N=20 THEN PRINT "A college education is a worthwhile and"
5730 IF N=20 THEN PRINT "stimulating experience"
5735 GOTO 305
5749
5750 for SIT
5755 IF N<10 THEN 5980 ELSE IF R=1 OR R=8 THEN R=9-R : GOTO 605
5760 PRINT "There's no CHAIR in here." : GOTO 305
5774
5775 for DRINK
5780 IF N<22 THEN 5980 ELSE IF L(4)=1 OR L(5)=1 THEN 5790
5785 PRINT "You're not carrying any MILK." : GOTO 305
5790 PRINT "Glugs; glugs." : L(4)=0 : L(3)=1 : GOTO 305
5799
5800 for HELP
5805 PRINT : PRINT "Lost? --- CONSULT MAP"
5810 PRINT "Not sure what you're carrying?---INVENTORY"
5815 PRINT "How many scoots are left?---CONSULT WATCH"
5820 PRINT "Want the room redescrbed?---RELOOK" : PRINT
5825 PRINT "What's a certain thing worth?---WORTH that thing"
5830 PRINT "Examples: WORTH COIN, WORTH CATALOG, WORTH CAT"
5835 PRINT : PRINT "Want a hint?---CONSULT a friend"
5840 PRINT "Examples: CONSULT BANKER, CONSULT " ; N$(29) ; PRINT
5845 PRINT "Not sure how to say something? --- LIST WORDS" : PRINT

```



```

5850 PRINT "Not sure what the clues mean? ---";
5855 PRINT "Think & experiment" : GOTO 305
5874 / (if you like) for you to fill in
5874 / for defaulting
5875 PRINT "The "; N$; " is hard to "; U$; ".": GOTO 305
5899 /
5900 /----- listing variables -----/
5902 /
5904 / Junk variables = I, J, A$, I9, I1, & a few others
5905 / H is Provincial to the 4300-4499 block.
5906 /
5908 / U9 = # of verbs understood
5910 / N9 = # of nouns understood
5912 / U = # for verb of command
5914 / N = # for noun of command
5916 /
5918 / M9 = # of manifestations
5920 / M = # of a manifestation
5922 / L(M) = location of manifestation M where
5924 / L(3) = 5 means object 3 is in room 5,
5926 / L(4) = -1 means object 4 is being carried by the Player,
5928 / L(6) = 0 means object 6 is in the inaccessible room
5930 / W(M) = worth of manifestation M (worth in points)
5932 /
5934 / W1( ) tells of N-S Passadeways
5936 / W2( ) tells of E-W Passadeways where
5938 / values of W1( ) & W2( ) have the following meanings:
5940 / 1 wall present (not really a passage at all)
5942 / 2 passage tried & clear; no demands or changes
5944 / 2.5 demands or changes, clear last time
5946 / 3.5 demands or changes, blocked last time
5948 / negative # means untried, trying makes Positive
5950 /
5952 / N W1(R)
5954 / M E rules handled by W2(R) (room R) W2(R+1)
5956 / S W1(R+R8)
5958 /
5960 / W1( ), W2( ), & W3( ) have to do with drawing the map.
5962 / W3(7) is an upper left corner -- see keypad for sense
5964 /
5966 / W4(R) = 1 if R has been visited & 0 otherwise
5968 / W4( ) used in naming rooms on the map
5970 /
5972 / R6 = # of floors in the house
5974 / R7 = # of rows of rooms on each floor
5976 / R8 = # of columns of rooms on each floor
5978 / R9 = # of rooms = R6*R7*R8
5980 / R0 = # of initial room = # of Bank room = R9-R8+1
5982 / R = # of current room
5984 /
5986 / P9 = # of Players (but Just one at a time)
5988 / P = # of the current Player
5990 / P$(P) = name of current Player
5992 /
5994 / T = total # of Passades made
5996 / T9 = limit on T
5998 / T1 = # Passades this turn of current Player
6000 / T2 = limit on T1
6002 /
6004 / Q9 = # questions asked in soliciting Player's choices
6006 / Q$( ) holds questions
6008 /
6010 / B9 = # of faces for the Banker
6012 / B$( ) holds faces of Banker
6014 /
6016 / C = # items carried including map but excluding watch
6018 / C9 = limit on C
6020 /
6022 / N1(M) = N means the N th noun describes manifestation M
6024 /
6026 / ESC$, VG$, NG$ are used only for Heath graphics
6028 /
6030 /----- listing of verbs, nouns, and manifestations -----/
6032 /
6034 / verb I noun I I manifestation I N1(I)
6036 /
6038 / TAKE N 1 catalog 20
6040 / GET S 2 coupon 13
6042 / GRAB E 3 empty bowl 12
6044 / PICK W 4 bowl with cold milk 12
6046 /
6048 / DROP NORTH 5 bowl with warm milk 12
6050 / GIVE SOUTH 6 carrot door 9
6052 / OFFER EAST 7 machine with bowls 0
6054 / DEPOSIT WEST 8 machine without bowls 0
6056 /
6058 / KICK DOOR 9 banker 23
6060 / CONSULT CHAIR 10 key 16
6062 / OPEN RABBIT 11 cat tellins of wanting milk 24
6064 / UNLOCK BOWL 12 cat with warm milk 24
6066 /
6068 / FILL COUPON 13 cat with cold milk 24
6070 / LIST WRAPPER 14 napping cat 24
6072 / WORTH MAP 15 usual rabbit 11
6074 / SIT KEY 16 rabbit giving egg 11
6076 /
6078 / DRINK EGG 17 rabbit eating carrot door 11
6080 / HELP COIN 18 initial rabbit promising 11
6082 / GO JEWEL 19 magic map 15
6084 / RELOOK CATALOG 20 wrapper 14
6086 /
6088 / URSE 21 voice giving reference 0
6090 / MILK 22 coin 18
6092 / BANKER 23 egg 17
6094 / CAT 24 Jewel 19
6096 /
6098 / WORDS 25 vase 21
6100 / INVENTORY 26
6102 / SCORE 27
6104 / WATCH 28
6106 /
6108 / CHULA 29
6110 / KLINKOVN 30
6112 / NELLAN 31
6114 /
6116 /----- end of Program -----/

```

# A Visit To The Enchanted House

*Nellan is Thirsty*

Let's pretend that you are in an enchanted house.

I will say what is happening to you and you should use one or two words to tell me what you want to do. I'll suggest messages at first.

If you have a question while you're inside the house, type HELP and perhaps I can help you.

When you have read all this, please

Push the button that says RETURN (or ENTER).

Are you the only person playing (YES or NO)? no

Please use capital letters.

Are you the only person playing (YES or NO)? NOPE

How many of you will be playing this time? 7

Please tell me your name

EVERETT Please give me the name of the next player.

TOMMY Thank you.

ALLISON Please give me the name of the next player.

ARMENTA Thank you.

B BOY Please give me the name of the next player.

ADAM Thank you.

LAUREN Please give me the name of the next player.

Thank you.



You are in a room called the Bank. A giant carpet on the floor says WELCOME EVERETT. A screen says that you are invited to visit for 72 scoots. 0 scoots have passed. To leave this room, type NORTH (or N or GO NORTH).

The BANKER, Mr KLINKOVN, looks happy.

There is a magic MAP here. If you GET the MAP, you may later CONSULT the MAP.

\*\*\* What do you choose to do?

GET MAP

Ok; you are now carrying the MAP.

The BANKER, Mr KLINKOVN, looks glad to see you.

\*\*\* What are you going to do now?

GO NORTH

Now you are in the White Room--the walls are white. There is a wall to the west. You know that the Bank is through the south doorway. There's a door to the north and a doorway to the east.

A white RABBIT is here. She hops to you and says, <<I am CHULA. Welcome to the Enchanted House, my friend, EVERETT>>. She hugs you and says, <<I have a present saved for you in the room to the east of here. If you say TAKE RABBIT (or TAKE CHULA) and next say GO EAST (or E or EAST), you can carry me to the Gold Room>>.

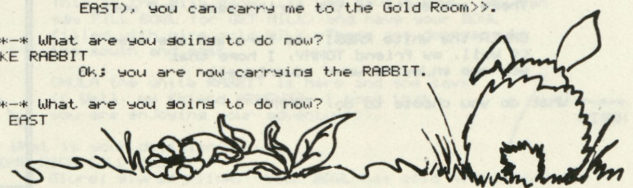
\*\*\* What are you going to do now?

TAKE RABBIT

Ok; you are now carrying the RABBIT.

\*\*\* What are you going to do now?

GO EAST



All the walls in this room are painted gold. There are exits to the north, east, and west.

CHULA hops out of your arms and scurries off. She returns and says <<Please TAKE this golden EGG that I hid here in the Gold Room>>.

A golden EGG is on the floor.

\*\*\* What now, EVERETT?

TAKE EGG

Congratulations, EVERETT. You'll earn points when the EGG is DEPOSITED (or DROPPED) in the Bank.

CHULA the white RABBIT is here and she says <<Well, my friend EVERETT, I hope that you are enjoying your adventure>>.

\*\*\* What is your decision?



DRINK EGG

The EGG is hard to DRINK.

\*\*\* What are you going to do now, EVERETT?  
CONSULT MAP

You are in the Gold Room now.

```
+++++ ? ? +++++
+++ ? +++++ ? +++++
+++ White Gold ? +++++
+++++ ++++++ ? +++++
+++ Bank ? +++++
+++++ ++++++ +++++
```

\*\*\* What do you choose to do, EVERETT?  
GO WEST

Now you are in the White Room--the walls are white.  
There is a wall to the west. You know that the Bank  
is through the south doorway. There's a door to  
the north and a doorway to the east.

\*\*\* What is your decision, EVERETT?  
GO SOUTH

You are in a room called the Bank. A giant carpet  
on the floor says WELCOME EVERETT.  
A screen says that you are invited to visit for 72  
scouts. 4 scouts have passed. To leave this  
room, type NORTH (or N or GO NORTH).

The BANKER, Mr KLINKOVN, looks glad to see you.

\*\*\* What are you going to do now?  
DEPOSIT EGG

Ok.  
The BANKER, Mr KLINKOVN, looks busy.  
Type SCORE to check your score.

A golden EGG is on the floor.

\*\*\* What is your decision, EVERETT?

SCORE  
You have 25 points out of 100.  
You earn points by DEPOSITING valuable things in the Bank.

\*\*\* What is your decision, EVERETT?  
NORTH

##-## It's TOMMY's turn now. ##-##

Now you are in the White Room--the walls are white.  
There is a wall to the west. You know that the Bank  
is through the south doorway. There's a door to  
the north and a doorway to the east.

\*\*\* What do you choose to do, TOMMY?

NORTH  
The door is locked and you're not carrying the key.

\*\*\* What are you going to do now?

GET KEY  
It seems that the KEY isn't here.

\*\*\* What now?  
GO EAST

All the walls in this room are painted gold.  
There are exits to the north, east, and west.

CHULA the white RABBIT is here and she says  
<< Well, my friend TOMMY, I hope that  
you are enjoying your adventure >>.

\*\*\* What do you choose to do, TOMMY?  
NORTH

This is the Milk Room. If you have a BOWL, you can  
see FILL BOWL (or GET MILK) and have your BOWL  
filled with nice cold MILK. There are doorways to  
the south and east.

An old man WRAPPER is on the floor.

\*\*\* What is your decision?

GET WRAPPER  
Ok; you are now carrying the WRAPPER.

\*\*\* What now, TOMMY?

GET MILK  
You need a BOWL.

\*\*\* What are you going to do now?

GET BOWL  
That's not possible now, TOMMY.



\*\*\* What are you going to do now, TOMMY?

GO EAST  
You cannot enter--the doorway is blocked by a thick  
growth of carrots.

\*\*\* What is your decision, TOMMY?  
GO SOUTH

All the walls in this room are painted gold.  
There are exits to the north, east, and west.

CHULA the white RABBIT is here and she says  
<< Well, my friend TOMMY, I hope that  
you are enjoying your adventure >>.

\*\*\* What do you choose to do?  
EAST

Oh it's hot in here. You are in the Hot Room and if  
cold MILK were in this room, it would be warmed.  
The MAP shows the doorways.

On the floor is a COUPON. According to what is  
written on the COUPON, if it is DROPPED in the Store  
while the machine is working, it will be good for one  
blue BOWL.

A voice says. <<Type HELP if you need help>>.

\*\*\* What is your decision, TOMMY?  
GET COUPON

Ok; you are now carrying the COUPON.

\*\*\* What are you going to do now?

EAST  
You can't walk through walls.

\*\*\* What now, TOMMY?  
CONSULT MAP

You are in the Hot Room now.

```
+++++ Milk +++++
+++ +++++ +++++
+++ White Gold Hot +++
+++++ ++++++ ? +++++
+++ Bank ? +++++
+++++ ++++++ +++++
```

\*\*\* What are you going to do now?  
SOUTH

##-## It's ALLISON's turn now. ##-##

You are in the room where NELLAN lives; this  
room called the Cat room.

NELLAN the CAT is here. She pretty-much fills up  
the room; she's as large as an elephant--no kidding.  
In a soft polite voice she says, << I'd love to have  
some nice cold MILK. Many of my friends have tried  
to deliver cold MILK here and many have failed.  
Perhaps you, ALLISON, would be kind enough  
to try. I must warn you: It will not be easy >>.

\*\*\* What is your decision, ALLISON?

PET CAT

I don't understand PET CAT. Perhaps  
you would like to type LIST WORDS to see a list of  
some of the words that I understand.

\*\*\* What do you choose to do?

LIST WORDS

I know these verbs: TAKE, GET, GRAB, OBTAIN, DROP, GIVE, OFFER, DEPOS  
IT, KICK, CONSULT, OPEN, UNLOCK, FILL, LIST, WORTH, SIT, DRINK, HELP, GO, RELOOK

I know these nouns: N, S, E, W, NORTH, SOUTH, EAST, WEST, DOOR, CHAIR  
BOWL, COUPON, WRAPPER, MAP, KEY, EGG, COIN, JEWEL, CATALOG, VASE, MILK  
CAT, WORDS, INVENTORY, SCORE, WATCH, CHULA, KLINKOVN, NELLAN.

I know some one word messages (for example: HELP)  
and I know some two word messages (for example:  
FILL BOWL, OFFER MILK). Whatever you say, I'll look  
at the first and last words; you could say  
SIT IN THE CHAIR or CONSULT THE SLEEPY MR KLINKOVN.

\*\*\* What is your decision, ALLISON?

GET CAT

The CAT is hard to GET.

\*\*\* What now?

WEST

You are standing in a room with green walls.  
A green CHAIR is in the corner.  
A sign here says:

```
-----
Masic works
in the
Enchanted House.
-----
```

A heavy COIN is here.

\*\*\* What are you going to do now?

GET COIN

Constratulations, ALLISON





You'll earn points when the COIN is DEPOSITED (or DROPPed) in the Bank.

\*\*\*- What do you choose to do, ALLISON?  
CONSULT MAP

You are in the Green Room now.

N  
|  
W---E  
|  
S

+++++  
+++ Milk ---  
+++++  
+++ White Gold Hot +++  
+++++  
+++ Bank +++ Green Cat +++  
+++++

\*\*\*- What do you choose to do?  
HELP

Lost? --- CONSULT MAP  
Not sure what you're carrying?---INVENTORY  
How many scoots are left?---CONSULT WATCH  
Want the room redescribed?---RELOOK  
What's a certain thing worth?---WORTH that thing  
Examples: WORTH COIN, WORTH CATALOG, WORTH CAT  
  
Want a hint?---CONSULT a friend  
Examples: CONSULT BANKER, CONSULT CHULA  
Not sure how to say something? --- LIST WORDS  
Not sure what the clues mean? --- think & experiment

\*\*\*- What is your decision?

WORTH WRAPPER  
The WRAPPER is worth 8 points.  
The WRAPPER might be useful as a tool.

\*\*\*- What now?

WORTH CAT  
The CAT is worth 8 points.  
The CAT might be useful as a tool.  
  
Friends are valuable.

\*\*\*- What is your decision, ALLISON?

CONSULT WATCH  
11 scoots have passed of your visit  
& 61 scoots remain.

\*\*\*- What are you going to do now, ALLISON?

CONSULT CAT  
It seems that the CAT isn't here.

\*\*\*- What are you going to do now, ALLISON?

E  
You are in the room where NELLAN lives; this room called the Cat room.

NELLAN the CAT is here. She pretty-much fills up the room; she's as large as an elephant--no kidding. In a soft polite voice she says, << I'd love to have some nice cold MILK. Many of my friends have tried to deliver cold MILK here and many have failed. Perhaps you, ALLISON, would be kind enough to try. I must warn you: It will not be easy >>.

\*\*\*- What are you going to do now?

CONSULT CAT  
The CAT laughs and says, <<Well, I know that you know a carrot-liking RABBIT>>.

\*\*\*- What is your decision, ALLISON?

THANK CAT  
I don't understand THANK CAT. Perhaps you would like to type LIST WORDS to see a list of some of the words that I understand.

\*\*\*- What do you choose to do?

N

Oh it's hot in here. You are in the Hot Room and if cold MILK were in this room, it would be warmed. The MAP shows the doorways.

\*\*\*- What are you going to do now?

W  
All the walls in this room are painted gold. There are exits to the north, east, and west.

CHULA the white RABBIT is here and she says << Well, my friend ALLISON, I hope that you are enjoying your adventure>>.

\*\*\*- What now, ALLISON?

GET CHULA  
You can only carry 4 objects (not counting the WATCH) and you are already carrying that much. If you really want to GET something else, then you will have to DROP something first. To see an inventory of what you're carrying, type INVENTORY.

CHULA the white RABBIT is here and she says << Well, my friend ALLISON, I hope that you are enjoying your adventure>>.

\*\*\*- What is your decision, ALLISON?

INVENTORY  
Your WATCH stays on your wrist for the whole visit. Here is an inventory of what you're carrying:  
COUPON  
MAP

WRAPPER  
COIN

\*\*\*- What do you choose to do, ALLISON?

DROP WRAPPER  
Ok.

CHULA the white RABBIT is here and she says << Well, my friend ALLISON, I hope that you are enjoying your adventure>>.

An old sum WRAPPER is on the floor.

\*\*\*- What is your decision?

GET CHULA  
Ok; you are now carrying the CHULA.

An old sum WRAPPER is on the floor.

\*\*\*- What now?

N

### It's ARMENTA's turn now. ###

This is the Milk Room. If you have a BOWL, you can say FILL BOWL (or GET MILK) and have your BOWL filled with nice cold MILK. There are doorways to the south and east.

CHULA hops down and sobbles a growth of carrots that was blocking the east doorway. She smiles and says, << Thanks, friend ARMENTA >>.

\*\*\*- What are you going to do now?

E

You are in the Store.

In a corner of the room is a machine which announces that if you DEPOSIT (or DROP) a COUPON and the machine is working, you will receive whatever the coupon is good for. If the machine doesn't work, you are invited to call the repairperson.

On the floor is a JEWEL.

\*\*\*- What do you choose to do, ARMENTA?

DEPOSIT COUPON  
Ok.

There is an clean empty blue BOWL here.

The machine in the corner has a flashing display announcing <<out of BOWLS>>.

On the floor is a JEWEL.

\*\*\*- What is your decision?

GET BOWL  
Ok; you are now carrying the BOWL.

The machine in the corner has a flashing display announcing <<out of BOWLS>>.

On the floor is a JEWEL.

\*\*\*- What now, ARMENTA?

GRAB JEWEL  
Congratulations, ARMENTA  
You'll earn points when the JEWEL is DEPOSITED (or DROPPed) in the Bank.

The machine in the corner has a flashing display announcing <<out of BOWLS>>.

\*\*\*- What now?

GO BACK TO THE WEST

This is the Milk Room. If you have a BOWL, you can say FILL BOWL (or GET MILK) and have your BOWL filled with nice cold MILK. There are doorways to the south and east.

CHULA the white RABBIT is here and she says << Well, my friend ARMENTA, I hope that you are enjoying your adventure>>.

\*\*\*- What is your decision?

GET SOME NICE COLD MILK  
Glugs; glugs; plink. Your BOWL has cold MILK now.

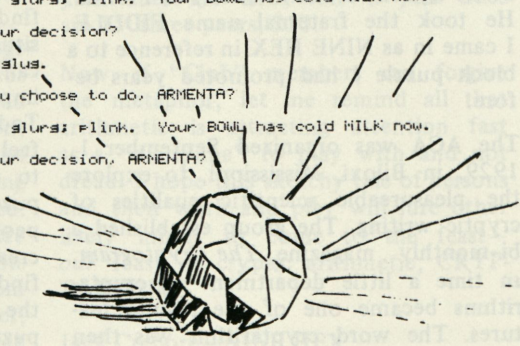
\*\*\*- What is your decision?

DRINK THE MILK  
Glugs; glugs.

\*\*\*- What do you choose to do, ARMENTA?

FILL THE BOWL  
Glugs; glugs; plink. Your BOWL has cold MILK now.

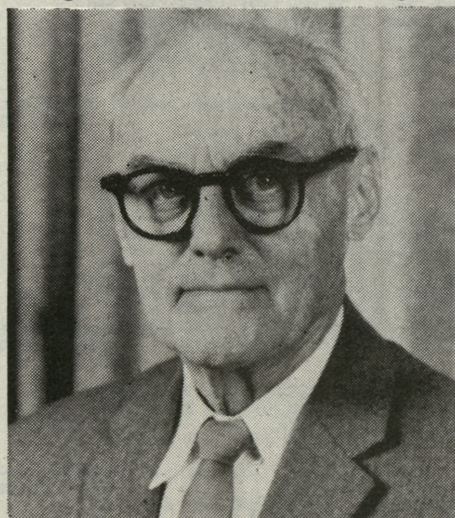
\*\*\*- What is your decision, ARMENTA?





# Cryptarithms From "Way Back"

BY JOHN DAVENPORT CREHORE



John Davenport Crehore, 1970

CRYPTARITHMS! Fascinating puzzles for fifty years! In what? In math. Oh... yes, *arithmetic* for all—kids to P.G.s! Ten letters for the ten digits; but which are these *two* digits:  $R \div R \div R : : TR$ ? Easy.

In the 1970s cryptarithms began to attract serious attention as stepping stones to *greatness* in mind power—yes, in reasoning power—both deductive and inductive reasoning, vastly different, not generally exercised together in formal education.

In the early 1930s a WWI flying pal of mine joined The American Cryptogram Association and the National Puzzlers' League. In 1936 he sponsored me for membership in both societies. Just now I am in only ACA. Let me introduce my WWI pal, with more to come. He is Col. F. D. Lynch, U.S.A.F. (Ret.) of Florida. He took the fraternal name FIDDLE. I came in as NINE HEX in reference to a block puzzle I had promoted years before.

The ACA was organized September 1, 1929, in Biloxi, Mississippi, to explore the pleasurable scientific qualities of cryptic writing. The group established a bi-monthly magazine, *The Cryptogram*. In time a little department of cryptarithms became one of the regular features. The word cryptarithm was then

colloquial but has finally been accepted and listed in *Webster's New International Dictionary*, 3rd Edition. The word was put together to describe arithmetic in hidden, cryptic form expressed in letters of the alphabet to designate the ten digits 0 to 9. The challenge is to determine each letter's digit value from its position and function in a problem.

There are now periodicals offering all sorts of mathematical puzzles: sticklers or just plain, simple, relaxing *fun* in twists that fascinate all ages. Cryptarithms abound. *Recreational Computing* is one of these periodicals. *RC's* introduction stressed "entertainment and home education." Another journal entitled "Cryptologia" is published by Albion College, Albion, Michigan 49224.

The authoritative, perspicacious *Wall Street Journal* published a front page article by Staff Reporter Liz Roman Callese, on March 13, 1978, headed "A Little Calculating And a Lot of Terror Equal Math Anxiety: Colleges and Others Attempt to Counter the Affliction: Tax Expert Hates Figures." The following is a quote from this article:

"If you're scared to death of mathematics, you are not alone. Listen: I can't do math and I don't want to do math, because it's uncomfortable.

Math anxiety, say a growing number of educators, is a severe ... problem ... afflicts women most ... keeping them from ... male-dominated fields and ... higher-paying jobs."

Students are flocking to colleges that have set up comprehensible elementary math courses. Persons who fear math find themselves handicapped even in situations designed to aid them. Many can't use pocket calculators. Executives impose arithmetic upon subordinates. Today there are millions of people who feel anxious about their scanty ability to do arithmetic. Why are crossword puzzles so popular? Because millions of people feel pleased that they have increased their education every time they find a wanted word. Cryptarithms offer the attractiveness, the fun, of crossword puzzles plus subconscious education in



Col. F.D. Lynch, U.S.A.F. (Ret.), 1970

arithmetic. Real fun in math education is new; cryptarithms are a delight to School Boards; cryptarithm-solving competition is already being played up in periodicals.

When I was admitted to the ACA, it was a great playing field of professional secret code users, from department store price labellers to international spies. Our national conventions, held jointly with our soro-fraternal society, the National Puzzlers' League, have always scintillated with intellect and risibles. I enjoyed greatly the one at Baltimore, Maryland, where I was sailing away in the spelling match until I missed that one word. Y'know?

WWII prevented attendance by those of our cryptanalysts who were in sensitive positions, but anyway we common folk frolicked on the green uh-h-h carpet if not -sward. We staged baffling charades, guessing games, spelling-backwards bees, extemporaneous orations, speed tests in ratiocination.

A little old lady (Aha! I fooled you!) seven-league boots, had travelled so widely we set her up as question answerer. Her instantaneous recall of facts and events stunned even us know-it-alls. She called herself a storehouse of useless information. We called her a marvel of mental keenness in old age.



One of our conventions, to give itself a taste of old times, went on an excursion on a Mississippi River paddlewheel, wood-burning steamboat.

We have now a real live little old lady who treasures a scroll awarded her by our societies for having solved 12,000 cryptic puzzles! She is Mrs. Dorothy K. Thomas, DOT. Failing eyesight and age (in the eighties?) have stopped her puzzle solving. But DOT says she enjoys her memories, living quietly in a little New Jersey town.

Dear NINE HEX;

I read and solved *most* of your cryptarithms... What intrigues me is your former association with the National Puzzlers' League.

I do believe I recognize your name from the old issues of *The Enigma* (NPL) that I have read, and am surprised and pleased to hear of you again...

NPL...hard times... '60s... early '70s... today thriving... Enclosed... program... The 140th Convention of the National Puzzlers' League, Stamford, Connecticut, July 20-22, 1979. ... names... new... fun... same...

Sincerely,  
Will Shortz (Willz)  
Associate Editor  
*Games Magazine*

There is certainly a strong appeal to teenagers in cryptarithms and other features of our two societies, ACA and NPL. I know of several who joined at age 13 or 14. One, now a Ph.D. in Computer Science, Dr. Frank Rubin (FIRE-O of ACA), joined at 14 and at 16 submitted an article that the Editor of *The Cryptogram* judged too technical for our membership! A later editor published it in 1975. That article treated solving of long multiplications by examining the end digits of the sub-products. I can get it from Cryptogram files, and expect permission from ACA to republish it in *RC* for our "Genius" solvers.

Dr. Rubin lives now in New York State. FIRE-O says a highlight of his young life was a convention in Washington, D.C., circa 1960. He won second prize in the major contest—among masters! A higher light shone when he was called upon to do technical writing on "di Crittografia," in the publication of a French edition of a world-famous Italian book. Degree Ph.D. in Computer Science did it!

If you ever become known as a multilingual technician, beware of Soviet secret agents in guise of novices asking inno-

cuous questions. Dr. Rubin was pestered for many months by phone calls and letters from ingeniously concealed sources, untraceable, becoming more and more specific about computer cryptography. Dr. Rubin concluded the man sought aid from unwitting amateurs, in breaking American cipher codes that Soviet professional spies hadn't been able to "crack." He believes the man is still at work. Nota bene, CIA, under your new name!

Another prominent member of ACA since 1949 has been U.S. Navy Captain Victor A. Moitoret (nom Je saurais) of New Mexico. He served as vice-president four years and then as president. He authored a notable pamphlet published by ACA: "Cipher Reference List and Index." His reminiscences emphasize that our two cryptic societies have active members in all 50 states and in 23 foreign countries—1974 data.

Capt. Moitoret points out that cryptography is an ideal diversion for persons restricted in movement. Equipment need be no more than pencil and paper, and textbooks for serious students. *Cryptarithms* provide the most fun clue-by-clue, as one solves the letters for the digits they represent in a problem. Here's a one-minute teaser for beginners:  $M \div M \div M :: KM$ , with only one digit that can represent M, and only one digit that can represent K.

I can invent others just as easy in seconds; harder ones in minutes. Problems whose solving requires genius, take hours to invent. Over the years I've accumulated hundreds of cryptarithms, unsorted although with solutions attached. Kids love 'em! "Oh! K'n I take this home? I want to see if Mother k'n do it!" (Why never Dad?)

Steady, Atlas! Earth is wobbling!

Cryptarithms are being published so widely that their devotees now may find them in all degrees of complexity. *Recreational Computing* features four problems in each issue, graded for Novice, Adept, Genius—and for Computer. *RC* has hundreds more on call, most of them fairly simple, all with clues or hints available for solution in pure logic with no random trial-and-error labor.

*The Cryptogram* presents a broad range of cryptography in which cryptarithms have only a small portion. But this broad range offers fascinating entertainment and practice to adepts. Je saurais has an ACA Certificate of Achievement citing 6,000 cryptic problems he has solved. In a newspaper article about him there is a reminder that police departments call upon ACA to decode messages from crank killers. Well, Capt. Moitoret, je saurais where to direct the police

if ever they call upon me. I shall refer them to you.

Near the beginning of this essay I introduced one of the brightest lights in our two soro-fraternities, ACA and NPL, retired Colonel Frederick D. Lynch, FIDDLE. He well exemplifies the intellectual heritage that all addicts to puzzling have, particularly puzzlers in mathematics, which calls for dual reasoning power—deductive and inductive.

Fiddle says that cryptarithms stimulate the imagination while training the mind. He was editor of the cryptarithm section of *The Cryptogram* for several years, and at other times contributed articles on cryptography. He observes that computers rob puzzling fans of the subconscious elation that comes from success in mental struggle.

Col Lynch is an honorary vice-president of ACA, the title recognizing his uncountable services to our societies. In 1964 he published an incomparably useful booklet for cryptarithm addicts—"An Approach to Cryptarithms." Its twenty-four 7" x 9" pages present hints, clues, procedures; tables of "everything" in the arithmetic book: units, squares, roots, factors up through decimals, un-, duo-, to quindecimals; detectable sequences; his "Negation Square" (my "Elimination Table" that I nicknamed for young people, "Is-not Chart").

Besides these useful, simple-to-comprehend aids, the booklet shows dozens of illustrations of situations that no one below a twelfth grader would really grasp! Altogether this is a tremendous, encyclopedic cryptarithm-solving reference book. ACA sells it (at least to members) at the nominal price of \$1.50. As minimum postage must be 48¢, I suggest that person wanting it send \$2.00 to Miss Isabel Murdock, Treasurer, The American Cryptogram Association, 1007 Montrose Avenue, Laurel, Maryland 20810.

The Association does not bid for members but would welcome the kind who would "join the Club" in spirit and perhaps action, but would not waste time in mere correspondence. Initiation and first year dues are \$10; current year dues are \$5; three years, \$10.

Now, if "Club" members can forgive the metaphor, let me remind all that arithmetic is attracting attention fast as a "discipline" to play with and not dread. I hope this sketchy tale of persons and their work and play will lure other lively minds to come to the feast—our feast of cryptic arithmetic, CRYPTARITHMS.

Jack Crehore—NINE HEX



# trs-80 art?

BY KEN HAHN

Tired of trying to use SET'S, RESET'S or PRINT'S to make your TRS-80 do something speedy on your screen, but they are just too slow? Then here's a fast one for you!

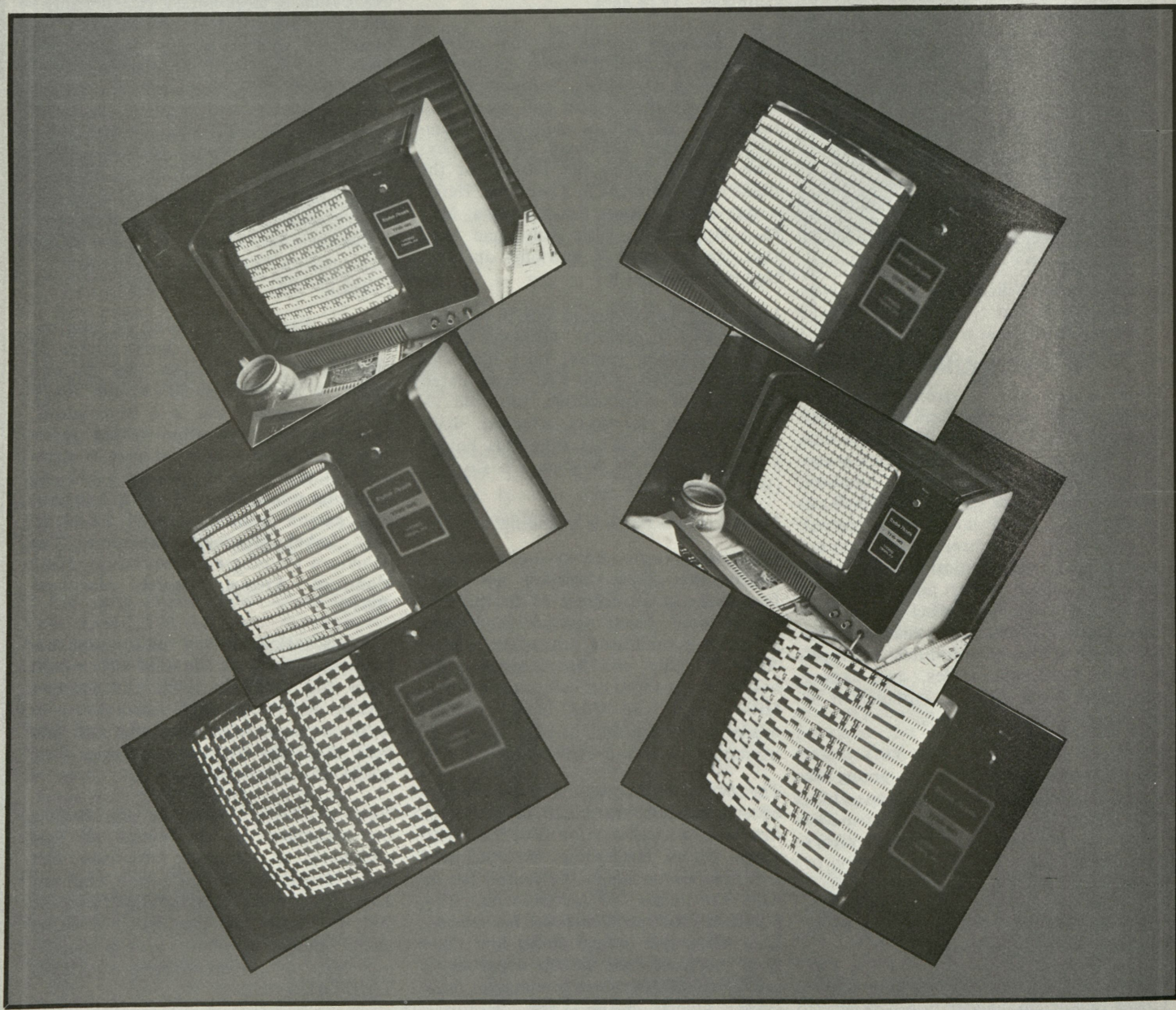
Here's the listing:

```
10 FORZ = 18000 TO 18013
20 READA:POKEZ,A
30 NEXTZ
40 DATA 33,0,56,17,0,60,1,0,4
50 DATA 237,176,195,80,70
60 SYSTEM
```

Run this and when a '\*\*?' comes up, type in '/18000' and the screen will fill up immediately. Try pushing in different keys, and the video will reflect the change in micro-seconds.

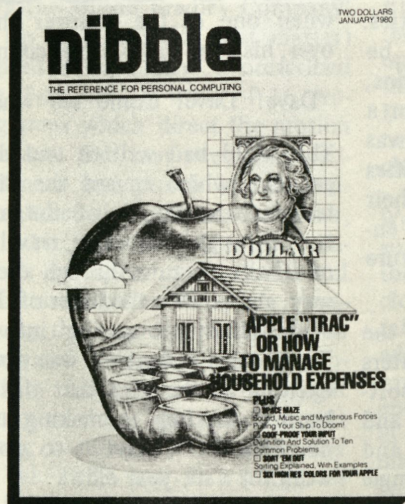
This program is a short machine language program, and it works by mapping out the keyboard memory on the screen. By changing the byte '56' on line 30, you can map out different portions of your RAM'S and ROM'S on the video.

Experiment by pushing in several keys at once with different combinations. For example, try 'DG349E' and watch it go!





# INTRODUCING . . . NIBBLE THE REFERENCE FOR APPLE COMPUTING



## **NIBBLE IS:**

A *SOFTWARE GUIDE* for high quality Applications Programs for your Home and Business.

## **NIBBLE IS:**

A *REFERENCE GUIDE* to new Programming Methods.

## **NIBBLE IS:**

A *BUYERS GUIDE* for making purchase decisions on new products.

## **NIBBLE IS:**

A *CONSTRUCTION PROJECT COOKBOOK* for adding function and value to the system you already own.

## **NIBBLE IS:**

A *COMMUNICATIONS CLEARING HOUSE* for users, vendors, and associations.

Each issue of NIBBLE features at least one significant new application program of commercial quality. The programs in NIBBLE are surrounded with articles which show how to USE the programming methods in your OWN programs.

Examples of upcoming articles:

- ☐ Modeling and Forecasting Your Business ☐ Build a Two-Tape Controller for \$12  
☐ Arcade Shooting Gallery — Save Your Quarters! ☐ Data Base Management  
 System I, II, III

And many many more! NIBBLE will literally "Nibble Away" at the mysteries of your system to help you USE IT MORE. In 1980, the principal featured system is the Apple II.

**Try a NIBBLE**

**nibble**

BOX 325 Lincoln, Mass. 01773

**I'll try NIBBLE !**

**Enclosed is my \$15 for 8 issues.**

☐ check ☐ money order

Name \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_

State \_\_\_\_\_ Zip \_\_\_\_\_



# COMMENTARIES

## MACHINES of LOVING GRACE

## - A CELEBRATION

BY DAVID THORNBURG

*What started out to be a community workshop on microcomputers turned out to be a stage debut for Big Trak, a mobile toy tank that is also a computer. In fact, Big Trak was the star of the show! — ED*

### A DAY AT THE COMPUTER ZOO

Several months ago (December 16th, to be exact), many people in the San Francisco Bay Area were doing their holiday shopping. Some people, however, had a much different experience that day—an experience that included Greek dancing, a pot-luck dinner, and the freedom to experiment with a large variety of personal computers. These people attended “Machines of Loving Grace,” a one-day workshop for “dragons and children of all ages.” The workshop, which takes its name from the poem “All Watched Over by Machines of Loving Grace” by Richard Brautigan, was co-sponsored by Dymax (with Bob Albrecht and Ramon Zamora) and by Innovision with the generous support and help of several people, including Liza Loop of the LO\*OP Center, Katie Thornburg of the Peninsula School Computer Project, and Dave Offen and Margie Rausch of the Peninsula School.

The workshop took place in the Peninsula School Auditorium—a large room in the old Coleman Mansion in Menlo Park. Tables were set up around the periphery of the room and a variety of computers was brought in: two Apples, two TRS-80's, five PET's, an Atari 800, and two Big Traks. Big Traks?—Ah yes, more on that later!

With the computer set up and tapes of Greek music playing in the background, we were all set for an experience that would be shaped by the desires of those who came. The object of this workshop was to provide the resources and let the

participants create their own activities. After all, since a computer could be turned into a world of caves and goblins, a space cruiser, a lemonade stand, or a splendid word game, our challenge was to provide a wide spectrum of activities for which people could make their selections and let the magic of the environment provide the cohesive structure for the day.

As the activities got started, many of the visitors gravitated towards the computers to sample the wealth of existing software we had on hand. As the clicks and beeps from Petball, SwordQuest and other popular games began to mingle in the air with the music and the laughter of the participants, Albrecht, Zamora and I became aware of another phenomenon taking place in the room. Several of the children (ages five to twelve or so) had discovered the Big Traks and were teaching themselves how to write programs. Big Trak, for the benefit of those of you who haven't seen one, is a large grey tank-like toy, actually a computer, that can be programmed to execute up to sixteen instructions in sequence. These instructions can cause the device to move forwards, backwards, turn left or right by various amounts, pause, and flash its lights. It contains a built-in keyboard connected to a TMS-1000 microprocessor and is programmed in a language similar to LOGO or SMALL-TALK.

Once one child had learned to program this gadget, he would share this knowledge with another child, until a whole crowd had mastered the rudiments of this device. Thus, with almost no adult intervention, several children mastered a computer programming language, learned how to de-bug programs, and were able to use this knowledge to create dances for these machines to perform.

For me, a highlight of this activity came

when one of the younger children ran over, his eyes wide with excitement.

“Dave! Dave! Come see what I did!”

This child had written and debugged a program which caused the Big Trak to traverse the length of the auditorium, turn through a doorway, travel the length of a hallway, turn through another doorway, and go into the foyer of the building—all without bumping into anything. To me this task itself was not nearly as exciting as the fact that this child had learned the key to making a computer do what he wanted it to do. Not bad work for a six-year old!

After dinner, it was finally time to shut the workshop down. One-by-one the computers were turned off and packed up and tables were put away. At last only one table was left with a PET and an Apple, each still getting heavy use. But by this time we had pretty well convinced the remaining enthusiasts that we would do this again soon. With the last table put away, the auditorium was once again quiet and the lights were turned off, bringing a most magical day to a close.

In his poem, Richard Brautigan says:

I like to think (and  
the sooner the better!)  
of a cybernetic meadow  
where mammals and computers  
live together in mutually  
programming harmony  
like pure water  
touching clear sky.

Well, Richard, for a few hours I think we did it!

(Note: The workshop described above was donated as a benefit for the Peninsula School Computer Project. Future workshops will benefit other projects as well, including Computertown, U.S.A.)





# IN DEFENSE OF HACKERS

BY STEVE KUDLAK

In one of my rarer optimistic moments I decided to produce my essay (as in the French word meaning "to try") in defense of hacking. I hope not to be too optimistic or ridiculous. We might entitle this one of the seven weird ideas of SKU; I don't know if it is worth distribution to large mailing lists, but I shall take my chances.

Weird idea number one: "Why is hacking so popular?" I feel that hacking is one of the few areas in modern life that comes close to the idea behind certain Renaissance guilds. For example, in general when people start hacking they learn something themselves, and when they encounter trouble they find an "adept" in the particular system or program and make inquiries getting information that is impossible to find in manuals or hard to find in documentation. During the course of this exchange, many working relationships and some personal relationships are formed.

The second guild aspect of hacking is that once you learn a very small set of things, you can DO something. It isn't like having a single person tie up several million pieces of equipment which he very well might damage. Have you ever tried to load an ultracentrifuge at 2 A.M. and not know if you are really balancing it right? If you do something wrong it gets wrecked, not just—oh well, my program died...

The other interesting thing I have seen is the WIDE variety of people communicating with one another. It is truly amazing. People who normally would not consider talking to each other end up conversing freely and even becoming friends. The other interesting aspect of the communication is that it causes a social levelling. I have seen Army colonel types learning from people twenty years their junior and barely out of high school.

The thing I find most encouraging about hackers is that on the whole the lot are COOPERATIVE rather than competitive. Maybe this is the result of being in a "protected" environment, but somehow I feel that even if this is true, the

cooperation is nonetheless valuable and should be encouraged. In a world where people can usually only be gathered together in cases of massive crisis or competitive games, this seems reassuring. The idea that another person wants to see your program succeed and is willing to help is pleasant and good.

In general some of the criticism of hackers being a narrow subculture is valid; however, I don't think the situation is worse than in other fields. For example, a hacker might worry about why the FOO JSYS returns something in AC2 or assumes everyone know that a AOBJN and a CAMLE are two different things operationally. This person is certainly no worse off in my mind than a person who worries about his "pects" and "tris" and "traps" and persists in struggling against the force of gravity to lift larger and larger metal objects into the air. Even funnier is that these people expect admiration and they often get it. Perhaps there are benefits from weight lifting, but certainly it is questionable whether lifting 70 Kg in the air is inherently any better than learning things to do with the CRJOB JSYS. (Please don't try to convince me that lifting metal objects in the air is really wonderful... I had a girlfriend who tried this for months. I could appreciate how she liked to do it, but I couldn't see anything wonderful in it myself.)

If there is one thing to be said about Americans it is that they LOVE to build, tinker and play around with devices. I think playing with technology is a good thing, and playing with technology in a supportive and cooperative environment is an even better idea. By playing with technology you can discover new uses for it.

The point is that technology can be a useful thing if people know its real as opposed to its advertised potential. Understanding it also makes it a LOT LESS FRIGHTENING. And loads of people get frightened by something they don't understand or haven't experienced.

In my opinion, people who know the potential of technology should try to increase technological literacy on a wide scale.

I don't think a technologically literate society is impossible. There are places where as much as 20% of the population is technologically literate. It can probably be achieved, provided we don't blow ourselves off the earth with a variety of political/nationalistic stupidities. The world would be much better off because of this enrichment.

you can tell  
it wabt to Do  
but you have  
to tell it just  
right  
this toy is  
Fun cuz it  
has a calculaton  
on it and I  
like it  
bigtrak  
bye Lisa

Big Trak review  
by Lisa Albers  
Age 6



# TOUCH PANELS AND INTERACTIVE GRAPHICS

BY JIM DAY

A touch panel is a device that senses the location at which it is touched by an object such as a fingertip. A number of schemes have been used to implement touch panels. Some use resistive membranes and some use light beams. The latter seem to be the most commonly used.

Typically, infrared emitting diodes and infrared sensing diodes are paired in a rectangular array mounted in front of a CRT screen. Electronic circuitry is used to sense which horizontal and vertical light beams are broken, passing this information to an associated computer. A computer program translates these data into X and Y coordinates of the touched location. This, in turn, allows applications programs to relate the touch of a finger (toe, nose, or whatever) to graphic or textual information being displayed on a specific area of the CRT.

This sounds simple enough and, in fact, it is simple and easily done. The combination of a touch panel and a CRT provides a very powerful tool for a wide variety of applications when used with interactive graphic displays. For example, instead of using a conventional electro-mechanical keyboard, one can simply display a picture of a keyboard on the CRT and input the keystrokes via the touch panel.

But using a touch panel to simulate a keyboard is a bit like designing an automobile in the shape of a horse. It can and has been done, but there are better ways of using a touch panel. Properly used, a touch panel provides a much friendlier interface than a keyboard. People who are intimidated by keyboards or feel that they will lose face by using a keyboard (traditionally a menial clerical function) are happy to use a touch panel. One example of how a touch panel can be used is in conjunction with a chess program. Instead of typing "D2-D4", one can simply touch the image of the man to be moved and then the image of the location to which it is to be moved. Touch panels can also be used with many other games such as Othello or checkers.

Touch panels are ideal for monitoring and controlling complex industrial processes such as oil refineries, assembly lines, and power plants. Instead of working with huge control panels with hundreds of buttons and flashing lights, one can use a single touch panel and CRT. Only those buttons actually needed at the moment need be displayed. Other buttons, text, and graphic indicators can be displayed as operating conditions change.

The same sort of thing can be done by the use of a "mouse" or a light pen. But a touch panel is a more convenient and natural interface. A computer equipped with a touch panel and CRT is ideal for a comprehensive residential command and control system. Various devices such as radios, television sets, telephones, and heating, cooling, and security circuits could be tied into the computer and controlled via the touch panel.

In the initial or "standby" mode the CRT would be blank, to avoid burning the screen with a constant picture. Or a constantly moving message, such as "Touch to view menu", could be displayed. A touch anywhere on the screen would cause a menu of primary functions to be presented (see Figure 1).

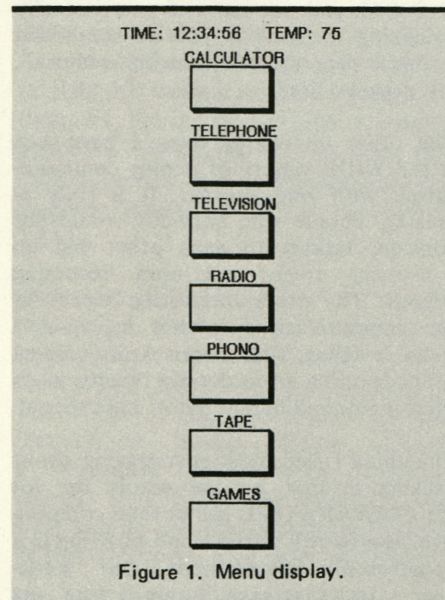


Figure 1. Menu display.

In the "menu" mode, the display could show the time and temperature and perhaps the current stock market report from a service such as Digicast. Below this display area would be a menu of touch-sensitive buttons for selection of various subsystems. The CALCULATOR button, for example, could display the buttons and standard readout of a desk calculator (see Figure 2). All of the buttons would be fully functional, thanks to the touch panel and, of course, the associated computer. Touching the MENU button would cause a return to the menu display.

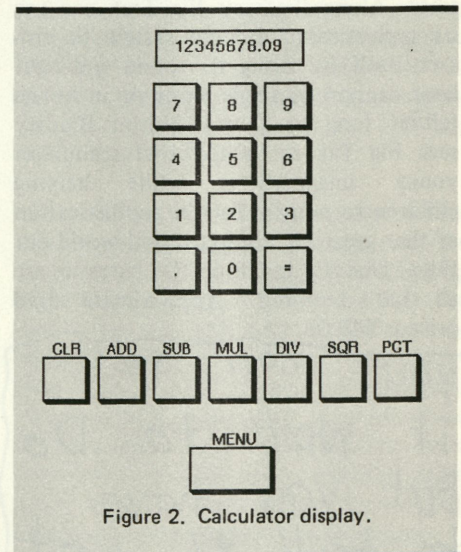


Figure 2. Calculator display.

Touching a TELEPHONE button on the menu could display a touch-tone keyboard, and a scrolling window above the keyboard could show a list of frequently called numbers (see Figure 3). An AUTO-DIAL button could dial a number selected by scrolling the display, or the touch-tone buttons could be used instead. During a telephone call, a videophone picture of the person called could be displayed on the CRT, assuming that such a service is available. At the bottom of the screen, a superimposed GOODBYE button could be used to end a call, or the computer could be programmed to respond to the spoken word "goodbye". A PLAYBACK button could be used to review calls made in, say, the last 24 hours, including those recorded while no one was at home.



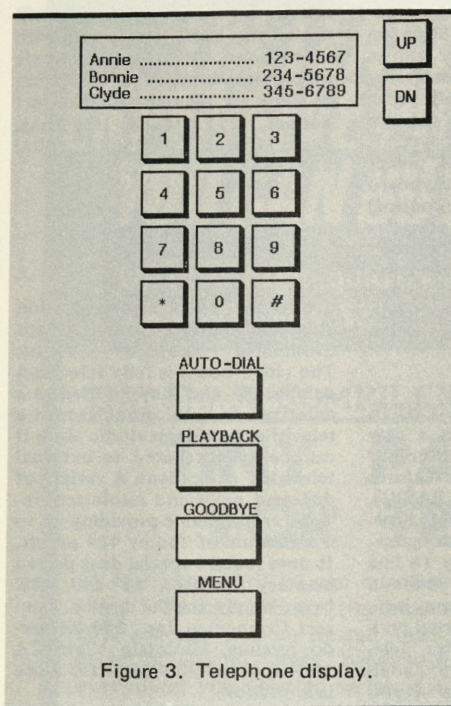
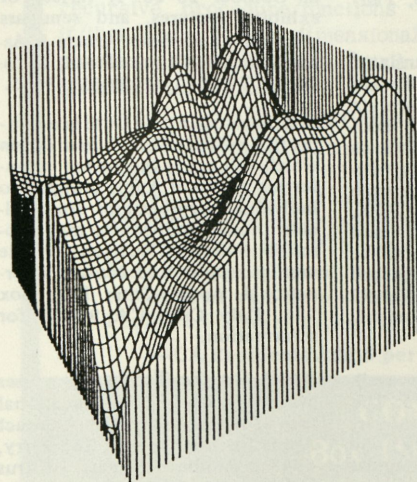


Figure 3. Telephone display.

Touching the TELEVISION button on the menu could display a monochrome or color TV picture with superimposed buttons and indicators for channel selection, fine tuning, etc. A better scheme would be to show the control buttons only if any part of the touch panel is touched. Otherwise the TV picture would fill the entire screen.

Likewise, a RADIO button could display all of the buttons and indicators required to operate an AM/FM radio, and PHONO and TAPE buttons could display controls appropriate to phonographs and tape decks.

The list of things that could be controlled by a computer and a touch panel is really endless, and I am sure that we are going to see many fascinating applications in the very near future.



## CLUB & CLASSROOM DISCOUNTS.

### PEOPLE'S COMPUTER COMPANY PUBLICATIONS

*Recreational Computing*  
*Dr. Dobb's Journal*

Clubs, users groups, schools and any others interested may take advantage of discounts for quantity orders of each of our magazines. See terms below.

All orders will be shipped via UPS—faster than the post office!

#### DISCOUNT SCHEDULE

(one year)

QUANTITY	DISCOUNT	RC	DDJ
Regular Subscription Rate/Year	none	\$12.00	\$19.00
5-10 copies	10%	10.80	17.10
11-24 copies	15%	10.20	16.15
25-49 copies	20%	9.60	15.20
50-74 copies	25%	9.00	14.25
75 + copies	30%	8.40	13.30

#### TERMS

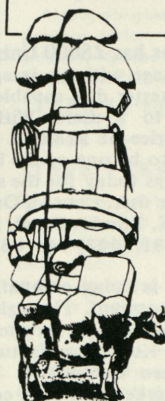
- Good in the U.S. only.
- Quantity orders must be made up of one title only—no mixing among titles.
- All copies must be shipped to same name and street address (no P. O. box).
- Payment must accompany order—check or money order only.
- Minimum one-year subscription.

#### Reader Service

## MOOVING?

#### IMPORTANT

Please advise us of address changes 60 days in advance and attach your magazine mailing label here. Please clip this notice and mail, or send us a reasonable facsimile.



Name (please print new information) \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_

State \_\_\_\_\_

Zip Code \_\_\_\_\_

Country \_\_\_\_\_

**Recreational  
COMPUTING**



P.O. Box E, 1263 El Camino, Menlo Park, CA 94025



# ANNOUNCEMENTS

## THIS PUBLICATION IS AVAILABLE IN MICROFORM



Please send me additional information.

### University Microfilms International

300 North Zeeb Road  
Dept. P.R.  
Ann Arbor, MI 48106  
U.S.A.

18 Bedford Row  
Dept. P.R.  
London, WC1R 4EJ  
England

Name \_\_\_\_\_  
Institution \_\_\_\_\_  
Street \_\_\_\_\_  
City \_\_\_\_\_  
State \_\_\_\_\_ Zip \_\_\_\_\_

## ADVERTISING SPACE AVAILABLE



1/4, 1/2, Full Page  
Inside  
Front & Back  
Covers

PLEASE SEND  
FOR RATE CARD

Advertising Manager  
People's Computer Company  
1263 El Camino Real, Box E  
Menlo Park, CA 94025  
(415) 323-3111

## Systems

The Apple III made its debut at NCC. It is a fully integrated computer system with built-in floppy disk drive, up to 128K bytes of memory, color and black & white video. The keyboard is new. It features a standard typewriter layout, a 13-key numeric pad, two user-programmable function keys, full upper-lower case character set (with a shift-lock that does the right thing for software that wants upper-case only), and auto-repeat on all keys. The system display features 80 columns.

Many previously optional features are now standard. The built-in floppy controller can handle up to three additional disks; the drives may be installed by plugging them into a connector provided. A printer port and an RS-232C serial port are standard. The system has a built-in clock and calendar maintained by their own internal battery to maintain the correct date and time even when the computer is turned off. A built-in 6-bit d/a converter can be used to synthesize simple music and voice.

The display includes both a text and graphic modes. Character matrices are stored in RAM so fonts may be reconfigured. In addition to 80-character text, a 40-character color-on-text mode and a 40-character Apple II emulation mode are provided. A black and white graphics mode (ultra-high resolution) provides a 560 x 192 grid. High resolution graphics with up to 16 colors is possible on a 280 x 192 grid.

With the system comes a new operating system, Pascal, BASIC, and FORTRAN. The operating system provides a uniform interface for all programs and makes the PEEK's and POKE's needed for many microcomputer systems unnecessary.

Onyx Systems has Z8000 Unix in a single package with an integral 8-inch Winchester disk capable of running up to 8 users. With a single-unit price of \$18,000 this package has to be one of the best computer buys today for the serious computer user. Contact Onyx Systems, 73 E. Trimble Road, San Jose, CA 95131, (408) 946-6330.

The PMC-80 is a plug-compatible TRS-80 replacement. Through an exclusive marketing agreement with a Hong Kong manufacturer, Personal Micro Computers, Inc. has begun marketing a plug-compatible version of the TRS-80 Model I, Level II. There are some enhancements: the cassette and power supply are built-in rather than separate units, the cassette interface is simpler and more robust, a built-in modulator allows a standard TV set to be used for the display as well as a TV moni-

tor. Even more interesting, in addition to the usual TRS-80 interface, a 50-pin connector allows interface to the S-100 bus. The PMC-80 is sold by mail order only. A system including cassette tape, 16K memory, Level II Microsoft BASIC in ROM, power supply, computer and keyboard (but excluding the TV monitor) was introduced at \$495; prices as of this writing are yet to be determined but will be competitive with the TRS-80. Contact Personal Micro Computers, 475 Ellis Street, Mountain View, CA 94043, (415) 968-1604.

The IMAGINATION MACHINE from APF Electronics is a new low-cost personal computer for home entertainment. It features user programmability (BASIC), 14K of ROM, 9K of RAM, typewriter style keyboard, two game-style controllers, a 32 by 16 line display format, color, a built-in cassette deck, microphone jack, and facilities to add external peripherals including a printer, telephone modem, mini-floppy, additional memory, and so forth. Graphics resolution is 128 x 192 with up to 8 colors. The suggested list price is \$599. For additional information contact APF Electronics, 444 Madison Avenue, New York, NY 10022, (800) 223-1264.

## Software

Stock Market Charting for the Apple II is possible with a \$99.95 technical analysis package available from RTR Software, P.O. Box 12351, El Paso, TX, (915) 544-4397.

The FICOMP Ham DX Package is a package of machine language programs and data which operates under 8080/Z80 CP/M systems to support the active radio amateur. The program computes specialized data (bearing, distance, time difference, etc.) from your site to a variety of foreign sites. Cost is \$22 postpaid from FICOMP, 3017 Talking Rock Drive, Fairfax, VA 22031.

A Structured BASIC Translator available from Acorn Software Products, 643 North Carolina Ave., S.E., Washington, D.C. 20003, (202) 544-4259, allows the user to write programs with structured control constructs such as PROCEDURE, CALL, CASE, IF-THEN-ELSE, WHILE, and UNTIL, and then translate the resultant program into standard BASIC. The program costs \$29.95 on disk and runs on the Model I, Level II TRS-80.

Pascal/M Release 2 is now available from Sorcim for \$175. The system consists of a compiler, emulator, runtime library and user manual. It operates under CP/M and supports full Pascal plus extensions (including ADA

syntax for random I/O). It will run on any 8080/Z80 system with or without AMD9511 arithmetic processors. An 8087/8088 version is soon to be released. Contact Sorcim, 2273 Calle De Luna, Santa Clara, CA 95053, (408) 988-3963.

## Hardware

Cromemco has a high-resolution graphics interface for use with Cromemco computer systems. The video board is fully television compatible and may be used as a colorizer or color modulator in a television broadcast studio since it can be synchronized to external television equipment. A variety of different color and resolution options are available providing up to a maximum of 756 by 484 points. It does require special dual-ported memory between 12 and 48K bytes simply for the display. Contact Cromemco Inc., 280 Bernardo Avenue, Mountain View, CA 94043, (415) 964-7400 for more information.

## Other

Dr. Dobb's Journal Volumes 1, 2, & 3 are now available in book form from Hayden as part of the People's Computer Company series. Contact Hayden Publishing Company, 50 Essex Street, Rochelle Park, NJ 07662.

All About Personal Computers is a 62-page report from Datapro covering 15 of the best-selling personal computers and including directories listing vendors of hardware, software, peripherals and publications. The price is \$25 from Datapro Research Corporation, 1805 Underwood Blvd., Delran, NJ 08075, (609) 764-0100.

Games Unlimited 1980 is a trade, consumer, and entertainment show scheduled for Hartford, CT, in October 23-26. A variety of exhibits, lectures, and seminars are planned. Contact Arnett & Associates, 25 Canton Road, Simsbury CT 06092, (203) 651-8631.

International Apple Core is a non-profit independent organization which will act as the parent organization for local Apple groups. No individual memberships are available though individuals may subscribe to the IAC quarterly. Apple user groups should contact International Apple Core, P.O. Box 976, Daly City, CA 94017 for information.

Apple for the Teacher is a user group specializing in educational uses of Apple Computers. Contact Apple for the Teacher, Ted Perry, 5848 Riddio Street, Citrus Heights, CA 95610 for information.



# THE LEAST EXPENSIVE PROGRAMS YOU CAN BUY.

Up to 77 high-quality programs for  
TRS-80, only \$8!

## THE \$15.50 COMPILER:

# TINY PASCAL FOR TRS-80

At last your TRS-80 can run Pascal too! The Chung/Yuen "tiny" Pascal is fully implemented for Level II TRS-80, 16K and up. You no longer need be left out of the growing group of Pascal users, because People's Pascal gives you everything you need to write structured Pascal programs:

- tiny Pascal compiler
- complete text editor for writing your programs
- complete tiny Pascal monitor
- sample Pascal programs
- user's manual (TRS-80 Computing issue 1:4)

People's Pascal is both a powerful, structured language and "CPU expeditor". People's Pascal programs execute at least four times faster than Basic, and often eight-times faster! Special functions open up the complete graphic capability of TRS-80. You now have the means to write those dazzling, impressive, high-speed graphics programs that are great for games, plotting, statistics, etc.

For the serious computerist, side two of People's Pascal II (tape 6) contains a larger compiler and complete source to the compiler, written in Pascal! This means you can re-compile the compiler, making changes, adding features, etc. (but this will take at least 36K RAM and a solid knowledge of programming).

With the complete People's Pascal operating system, you can save and load both source (Pascal) programs, and compiled programs, to or from cassette tape. This means that once you have de-bugged a program, you can save the P-code (compiled program) and thereafter, to run the program, you need only load the super-fast P-code.

Here is a partial list of People's Pascal features: recursive procedure/functions • for (loop) • case if/then/else • one-dimensional arrays • write • read constant • repeat/until (loop) • "peek & poke" • plot (graphics for TRS-80)

### DEALER INQUIRIES INVITED

People's Pascal 1 (tape 3) is written in Basic, implemented for TRS-80 by John Alexander of Berwick Australia. It compiles P codes more slowly and is harder to use than Pascal 2, but its P codes can be translated into Z80 native code and saved as System tapes. Pascal 2 requires that Pascal be resident at run time—Pascal 1 does not. Other People's Software tapes \$8.

—CA Residents Add 6 per cent TAX (\$8.45, \$16.40 & 24.88). Quoted Prices INCLUDE \$.50 Shipping—

## TAPE 1 LEVEL 2

Mortgage calculations, Dow Jones Industrial, cash flow, inventory-change, California income tax, journal ledger (8K), loan amortization, perpetual calendar, bio rhythm, payroll, diet planning, speed reading, touch typing, sales receipt tally, decision maker, mail addressing, straight depreciation, double-declining depreciation, and revolving charge account.

Also, math problems, queen, Star Trek I, number guessing, wheel of fortune, World War II bomber, rock-scissors-paper, seek, Star Trek II, Red Baron, mini-Trek, strategy, pilot, battleship, "On A Snowy Evening", mastermind, tic-tac-toe, grand prix auto race, capitals, etch sketch, hangman. Total programs: 34; Level 1 version available: 24 programs. \$8

## TAPE 2 Some Common Basic Programs (lev. 2)

Fully documented in *Some Common Basic Programs* by Lon Poole & Mary Borchers (Osborne & Associates, 630 Bancroft way, Berkeley CA 94710—or from CIE—\$12.50 postpaid from CIE, via UPS, CA residents add tax (to \$13.25)):

Investment, future value regular deposits, regular deposits; regular withdrawals, initial, minim (for withdrawals); nominal interest, effective & earned-interest; depreciation rate, amount depreciation; salvage value; discount com'l paper; loan principal, regular & last payment, remaining balance, term-loan; mortgage amortization; greatest common denom. integer prime factors; polygon area; triangle parts; analysis, operations two vectors; radian-degree, degree-radian conversion; coordinate, polar equation, functions plot; linear, curvilinear interpolation; Simpson's & trapezoidal rules, Gaussian quadrature integration; derivative.

Side 2—quadratic equation, polynomial (Newton) & half-interval-search roots; trig polynomial; simultaneous equations; linear programming; matrix addition, subtraction; scalar multiplication, inversion; permutations & combinations; Mann-Whitney U test; mean, variance, standard deviation; geometric mean & deviation; binomial, Poisson, normal, Chi-square distribution; Chi-sq., student's T-distribution test; F-distribution; linear correlation coefficient; linear, multiple-linear, Nth order, geometric, exponential regression; system reliability; future projections; Federal withholding taxes; tax depreciation schedule; check writer; recipe cost; map check; day of week; days between two dates; anglo to metric; alphabetize. \$8

## TAPE 3 People's Pascal Development System 1

Pascal 1 compiler-program development system. \$15.50

## TAPE 4 LEVEL 1

Election returns, business percentage, ups and downs of business, index, inventory control, sales receipt tally, gas mileage, driving distance, mixed monthly sales report, payroll, annual earnings, speech recording aid, and double-declining depreciation.

Also, math problems, cash register, chase, snoopy, commander-in-chief, Christmas graphic, air raid, balance scale, stock market, tic-tac-toe and On A Snowy Evening. \$8

## TAPE 5 LEVEL 2

Memory test, mortgage payments, tension breaker, lineprinter-screen & vice-versa utilities, Federal income tax, election returns, business percentage, vacation planner, car pool (disk), diet planning 2, mailing list (disk) and first aid.

Also spelling bee, Star Trek 3, mind bender, tachistoscope, chase, common factor, klingon capture, spelling practice, Hamurabi, animals, Snoopy, cryptogram, starship, ants, Yesterday, and Pilot (disk). Pilot is the language of computer-aided instruction (CAI)..

## TAPE 6 People's Pascal 2

Pascal 2 compiler-program development system. \$23.50

## TAPE 7 LEVEL 2

Disassembler, Pilot, roster, dropout, memory loader, memory sort, inventory control, graph, land surveying, mixed monthly sales report, shopping list, diet planning 3, loan progress chart, hex-decimal conversion.

Also Star Trek 4, states and capitals, battleships 2, spelling practice 2, number guessing, hangman 2, snark, slot machine, cipher, target, surround, adder, termites, lunar lander, multiplication exercise, five-in-a-row, Bastem, and write. A number after a program indicates there are other similar People's Software programs. Pilot is the same as the disk pilot on tape 5, except runs on 16K tape systems. \$8

**COMPUTER INFORMATION EXCHANGE**  
Box 159, San Luis Rey CA 92068 (714) 757-4849



# A Proposed Graphics Language

BY JIM DAY

The display of high-resolution graphics is possible on many kinds of small computers. This is often done by adding special graphics commands to some popular language such as Basic. Graphics commands are generally rather primitive, allowing one to plot points, draw straight lines, and display predefined patterns. If an interpreter is used rather than a compiler, the speed of execution is usually too slow for satisfactory animation.

Because it takes time to change a graphic display, it's a good idea to alternate between two display areas in memory, so the old picture will remain on the screen until the new picture is ready for display. Another way of hiding display updates from the viewer is to blank the screen until the new picture is ready. This may produce flicker in the display but has the advantage of freeing the computer from the task of maintaining the display on the screen, which can easily consume more than half of the computer's time.

Suppose we had a really good graphics language at our disposal. Let's call it Really Advanced Pictorial Image Display, or RAPID for short. Let's assume that this is an interactive language that allows pictures to be defined manually via a mouse, stylus, or joystick as well as by source code. Let's further assume that the language processor is smart enough to translate a manually entered picture into the corresponding source code. It should also be possible to edit a picture online and have the processor modify the source code accordingly.

Ok, so what kind of graphics statements do we want RAPID to understand? To define, but not display, a dot at screen location X,Y we would like to write:

DOT FOO IS X,Y

FOO is the symbolic name assigned to the dot. To actually display the dot after it has been defined, we will use the statement:

DRAW FOO

In real life, one seldom wants to display a single isolated dot. Usually one deals with lines and groups of lines. So we need a line defining instruction:

LINE ZAP IS X1,Y1 TO X2,Y2

This defines a straight line, named ZAP, running from screen location X1,Y1 to location X2,Y2. Naturally, we expect RAPID to complain if either endpoint lies off the screen. If the display device uses raster scanning, most lines won't be perfectly straight but will be approximated by a series of line segments that are either horizontal or vertical. But let's not worry about that. To relocate line ZAP, we can write:

SHIFT ZAP BY X,Y

X and Y specify horizontal and vertical bias values for line ZAP. The length and direction of the line are unchanged. To rotate line ZAP about some point, not necessarily on the line itself, we can write:

ROTATE ZAP ABOUT X,Y BY A

X,Y defines the point about which ZAP is to be rotated and A is the angle of rotation, expressed in degrees. Rotation will be counterclockwise for positive values of A. Nothing happens on the screen until a DRAW instruction is executed:

DRAW ZAP

It would be nice to define a group of lines forming some figure such as a rectangle:

FIGURE FIG1 IS LIN1,LIN2,LIN3,LIN4

The arguments LIN1, LIN2, LIN3, and LIN4 are the names of previously defined lines. FIG1 is the name given to the figure comprising the four lines. We can then use FIG1 in other statements such as:

ROTATE FIG1 ABOUT X,Y BY A

An obvious extension of this is to allow figures to be composed of previously defined figures and/or lines:

FIGURE FIG2 IS FIG1,LIN5,LIN6

FIG2 is composed of FIG1 plus lines LIN5 and LIN6. To duplicate the shape of an existing figure, we can write:

FIG9 IS FIG7

The location of FIG9 can then be adjusted via a SHIFT or ROTATE statement. It would be useful to define circles:

CIRCLE C1 IS R AT X,Y

C1 is defined as a circle of radius R whose center is at location X,Y. Circular arcs are also useful:

ARC ZOT IS R AT X,Y FROM A1 TO A2

ZOT is a circular arc of radius R whose center is at X,Y. It extends from angle A1 to angle A2. To remove graphic elements from the screen we need a statement to do the inverse of what DRAW does:

ERASE FIG3,LIN4

This will erase all elements of FIG3 as well as LIN4. An ERASE statement with no arguments will erase the entire screen. If we want to make an element larger or smaller without changing the shape, we can use a SCALE statement:

SCALE FIG7 BY 2

This will double the size of FIG7, using the center of the figure as a fixed reference point. There should be a way of defining text strings:

TEXT TXT1 IS "FRONT VIEW" AT X,Y

This defines a text element named TXT1 comprising the string "FRONT VIEW" and having a starting location of X,Y. Once defined, a text element can be displayed via a DRAW statement or can be included as part of a figure:

FIGURE FIG7 IS FIG6,TXT1

If a figure is defined as including text, the text can be shifted, scaled, or rotated along with the rest of the figure:

SHIFT FIG7 BY X,Y

Animation can be accomplished by program loops containing appropriate DRAW, ERASE, and element redefinition statements.



# BACK ISSUES

## PCC NEWSPAPERS

### Half Off Sale!

All 12 available back issues of *PEOPLE'S COMPUTER COMPANY NEWSPAPER* for only \$12.

- #3 May-June 1973 ZINGO/"Going to Bid"/How to Buy an Edusystem/BASIC MUSIC/CHOMP. \$2.00
- #10 July-Aug 1974 A Fantasy of Future Forms/Low-Cost Home School Microprocessor Systems/INCHWORM/MAZE/Analysis of Teaser/Computer Illiteracy Problem by Peter Lykos. \$2.00
- #13 Jan-Feb 1975 Build your own BASIC/Mysterious RND/More INCHWORM/Computers & Art/BIOSIN Son of Hardware. \$2.00
- #18 Nov-Dec 1975 NUMBER/Tiny BASIC/Analog to Digital Conversion/Minuet in G Major. \$2.00
- #19 Jan-Feb 1976 Space Games/Tiny BASIC Soloworks/DIDDLE/STTR/More Music. \$2.00
- #20 Mar-Apr 1976 Games for the Pocket Calculator/SQUARES/Make Believe Computers/STRTRK/Huntington Computer Project. \$2.00
- #21 May-June 1976 Biofeedback/POUNCE/"We're Building Our Own Computer"/Programmer's Toolbox/CCC/TV as Terminal and Game Center. \$2.00
- #22 July-Aug 1976 Low Cost Software/Tiny BASIC, Tiny Trek/Your Brain is a Hologram. \$2.00
- #23 Sept-Oct 1976 Dungeons & Dragons/HATS/One on One/PLANETS/The Positive of Power Thinking. \$2.00
- #24 Nov-Dec 1976 STORY/SNAKE/More Build Your Own Computers/Introducing PILOT/FROGS. \$2.00
- #25 Jan-Feb 1977 REVERSE/Robots/Tiny PILOT/Space & Computers/Conversational Programming. \$2.00
- #26 Mar-Apr 1977 Z-80 PILOT/6502 Assembly Programming/Tiny BASIC for Beginners. \$2.00

Only the issues listed above are available. Price includes issue, handling, and shipment by second class or foreign surface mail. Within the U.S., please allow 6-9 weeks to process your order second class. Outside the U.S., surface mail can take 2-4 months. For faster service within the U.S. we'll ship UPS if you add \$1.00 for 1-2 issues and \$.50 for *each* issue thereafter. We need a street address, not a PO Box. Outside the U.S., add \$1.50 *per* issue requested for airmail shipment.

## PEOPLE'S COMPUTERS

- #27 May-June 1977 Graphics, Computers & Copyright Laws, PILOT & Tiny BASIC, Home Computers for Beginners, Women & Computers. \$2.50
- #28 July-August 1977 Heathkit's 8080 & LSI-11, Do-It-Yourself CAI, Robot PETs. \$2.50
- #29 Sept-Oct 1977 Computer Networks, The \$595 PET, More Tiny Languages. \$2.50
- #30 Nov-Dec 1977 Oz Graphics, Bio-feedback & Micros, Our PET's First Steps. \$2.50
- #31 Jan-Feb 1978 Robots, Video Discs, PET Update & Programs, PASCAL & COMAL. \$2.50
- #32 March-Apr 1978 Epic Computer Games, TRS-80 Review, Micros for the Handicapped. \$2.50
- #34 July-Aug 1978 Computer Whiz Kids, Public Access to Computers, Man-made Minds. \$2.50
- #35 Sept-Oct 1978 Computer & Museums, Kingdom, APL, Sorcerer of Exidy. \$2.50
- #36 Nov-Dec 1978 The Return of the Dragons, APL, Animated Games for TRS-80. \$3.00

## RECREATIONAL COMPUTING

- #37 Jan-Feb 1979 Artificial Intelligence, A Fantasy by Jules Verne, TRS-80 Personal Software, The Apple Corps is with Us. \$3.00
- #38 March-Apr 1979 Artificial Intelligence, Calculator Comics, Chess Reconsidered, Lord of the Rings. \$3.00
- #39 May-June 1979 Now Peter Can Read, Game of Life, TRS-80 Programs, Detective Game Program. \$3.00
- #40 July-Aug 1979 Computertown, U.S.A.I., Cryptarithms, DP Games Playing, FORTE Music Programming Language. \$3.00
- #42 Nov-Dec 1979 Shogi Games, Creating New Characters on the Ti 99/4, Music for Atari, TRS-80 & PET. \$3.00
- #43 Jan-Feb 1980 Computing for Health and Equality (PET), Ti Graphics & Animation Pt. 2, More Music for Atari. \$3.00
- #44 Mar-Apr 1980 Special Games Issue! Recreation Apple II Hi-Res Graphics, Galaxy II, TRS-80 Tiny Trek. \$3.00
- #45 May-June 1980 Intro to Computer Music, CBBS Phone Numbers, 6502 Machine Language, Five New Personal Computers. \$3.00

Circle your choices.

3 10 13 18 19 20 21 22 23 24 25 26 27 28 29 30  
31 32 34 35 36 37 38 39 40 42 43 44 45

Please send me the issues circled.

☐ Yes, send me #3-26 for only \$12

☐ I'm ordering \$20 or more and I want a 10% discount.

I enclose \$ \_\_\_\_\_ (U.S. check or money order). Foreign add \$.50 *per* issue (this includes postage and handling).

Please send in this form or facsimile to: Recreational Computing, 1263 El Camino Real, Box E, Menlo Park CA 94025.

My Name \_\_\_\_\_

Address \_\_\_\_\_

City/State/Zip \_\_\_\_\_





# PROGRAMMING PROBLEMS

© 1980, BY BOB ALBRECHT, DON ALBERS AND JIM CONLAN

## PROBLEM #19 WORD'S WORTH NUMBER TWO

You did do PROBLEM #11 WORD'S WORTH NUMBER ONE, didn't you? Of course you did. Oh...you didn't? Well, never mind. If you need it to help you in this one, skip back and do it.

Remember: A = 1, B = 2, C = 3, and so on up to Z = 26. The worth of a word is the sum of the number values for the letters in the word.

Here is a list of the first 100 prime numbers.

2	3	5	7	11	13	17	19	23	29
31	37	41	43	47	53	59	61	67	71
73	79	83	89	97	101	103	107	109	113
127	131	137	139	149	151	157	163	167	173
179	181	191	193	197	199	211	223	227	229
233	239	241	251	257	263	269	271	277	281
283	293	307	311	313	317	331	337	347	349
353	359	367	373	379	383	389	397	401	409
419	421	431	433	439	443	449	457	461	463
467	479	487	491	499	503	509	521	523	541

For each prime number, find a word whose word's worth is equal to the prime number. We will accept any single unabreviated word in *Webster's Third New International Dictionary* (Unabridged). Good hunting!

## PROBLEM #20 SUM AND PRODUCT OF DIGITS OF A TWO-DIGIT NUMBER

Write a program to compute and print or display the sum of the digits and the product of the digits of a two-digit positive integer. A two-digit positive integer is:

- (1) an integer,
- (2) greater than or equal to 10,  $10 \leq X \leq 99$
- (3) less than or equal to 99.

Sample RUN.

TWO-DIGIT POSITIVE INTEGER? 37  
SUM OF DIGITS = 10  
PRODUCT OF DIGITS = 21

TWO-DIGIT POSITIVE INTEGER? -37  
OOPS! ENTER AN INTEGER, 10 TO 99.

TWO-DIGIT POSITIVE INTEGER? 7.3  
OOPS! ENTER AN INTEGER, 10 TO 99.

TWO-DIGIT POSITIVE INTEGER? 02  
OOPS! ENTER AN INTEGER, 10 TO 99.

TWO-DIGIT POSITIVE INTEGER? and so on.

Accept *only* integers in the range 10 to 99, inclusive. Try some of these to test your program.

TWO-DIGIT INTEGER	SUM OF DIGITS	PRODUCT OF DIGITS
10	1	0
28	10	16
99	18	81

## PROBLEM #21 SUM AND PRODUCT OF DIGITS OF A THREE-DIGIT NUMBER

If you are a novice programmer, you may wish to first do PROBLEM #20, then this one.

Write a program to compute and print or display the sum of the digits and the product of the digits of a three-digit positive integer. A three-digit positive integer is:

- (1) an integer
- (2) greater than or equal to 100,  $100 \leq X \leq 999$
- (3) less than or equal to 999.

Sample RUN.

THREE-DIGIT POSITIVE INTEGER? 123  
SUM OF DIGITS = 6  
PRODUCT OF DIGITS = 6

THREE-DIGIT POSITIVE INTEGER? 99  
OOPS! ENTER AN INTEGER, 100 TO 999.

THREE-DIGIT POSITIVE INTEGER? 1234  
OOPS! ENTER AN INTEGER, 100 TO 999.

THREE-DIGIT POSITIVE INTEGER? 3.14  
OOPS! ENTER AN INTEGER, 100 TO 999.

THREE-DIGIT POSITIVE INTEGER? 007  
OOPS! ENTER AN INTEGER, 100 TO 999.

THREE-DIGIT POSITIVE INTEGER? and so on

Accept *only* integers in the range 100 to 999, inclusive. Try some of these to test your program.

THREE-DIGIT INTEGER	SUM OF DIGITS	PRODUCT OF DIGITS
100	1	0
394	16	108
999	27	729



## PROBLEM #22 TWO-DIGIT NIVEN NUMBERS

Niven numbers? Until recently, we had never heard of them. Then, the Winter 1980 issue of the MATYC JOURNAL\* arrived. The cover story was "Mathematical Discovery and Niven Numbers" by Robert E. Kennedy, Terry A. Goodman and Clarence H. Best.

A Niven number is a positive integer which is divisible by the sum of its digits.

- 24 is a Niven number because  $2 + 4 = 6$  is a divisor of 24.
- 37 is *not* a Niven number because  $3 + 7 = 10$  is *not* a divisor of 37.

Write a program to compute and print or display all two-digit Niven numbers. We want all Niven Numbers in the range 10 to 99, inclusive.

For more info on Niven numbers, contact Robert E. Kennedy, Mathematics Dept., Central Missouri State University, Warrensburg, MO 64093.

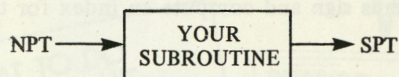
\*THE MATYC JOURNAL is published three times a year by the Department of Mathematics/Statistics/Computer processing, Nassau Community College, Garden City, NY 11530. One year (3 issues) for \$8.50.

## PROBLEM #23 SUM OF DIGITS SUBROUTINE

Write a subroutine to compute the sum of the digits of a positive integer.

NPT = a positive integer.  
SPT = sum of the digits of NPT.

If NPT = 235, then SPT =  $2 + 3 + 5 = 10$



Use this program, or a similar program, to check out your subroutine.

```
100 REM *** SUM OF DIGITS OF A POSITIVE INTEGER
200 REM *** ASK FOR AN INTEGER
210 CLS
220 PRINT : INPUT "ENTER AN INTEGER, 1 TO 999999" ; N
230 IF N <> INT(N) OR N < 0 OR N > 999999 THEN 220

300 REM *** USE SUM OF DIGITS SUBROUTINE
310 NPT = N
320 GOSUB _____ line number of your subroutine.

400 REM *** PRINT SUM OF DIGITS
410 PRINT "THE SUM OF DIGITS OF " NPT "IS" SPT
420 GOTO 220
```

Your SUBROUTINE

In the above program, we limit numbers to integers in the range 1 to 999999 (line 230). Change this limit if you wish.

In the January/February, March/April, and May/June issues of *Recreational Computing*, the "Programming Problems and Solutions" series by Bob Albrecht, Don Albers, and Jim Conlan should have included the notation "Copyright 1980 by Bob Albrecht, Don Albers, and Jim Conlan" as they retain personal copyright to the material.

# SOLUTIONS TO PROBLEM #1

BY BOB ALBRECHT, DON ALBERS AND JIM CONLAN

The "Programming Problems" section of *RC* began in the Feb-Mar 1980 issue with the following problem.

## PROBLEM #1 POSITIVE, NEGATIVE OR ZERO

An easy problem. All we want is a program that asks for a number, then tells you something about the number.

- If you enter a positive number, the computer tells you:  
YOUR NUMBER IS POSITIVE
- If you enter a negative number, the computer tells you:  
YOUR NUMBER IS NEGATIVE
- If you enter the number zero (0), the computer tells you:  
YOUR NUMBER IS ZERO

A solution to this problem is shown below, in TRS-80 BASIC, a version of Microsoft<sup>TM</sup>BASIC. With minor editing it will run on the APPLE, ATARI, PET and several other computers.

```
100 REM***PROBLEM #1 POSITIVE, NEGATIVE OR ZERO
110 REM***RECREATIONAL COMPUTING, JAN/FEB 1980
120 CLS

300 REM***ASK FOR A NUMBER, X
310 PRINT : INPUT "NUMBER, PLEASE" ; X

500 REM***TELL WHETHER NUMBER IS POSITIVE,
    NEGATIVE OR ZERO
510 IF X > 0 THEN PRINT "YOUR NUMBER IS POSITIVE"
520 IF X < 0 THEN PRINT "YOUR NUMBER IS NEGATIVE"
530 IF X = 0 THEN PRINT "YOUR NUMBER IS ZERO"

700 REM***GOTO 'ASK FOR A NUMBER, X'
710 GOTO 310
```

Ha! We have pre-empted the solution which immediately leaped to your mind. In fact, you *can't* use the IF statement at all! *Your* Program must "behave" exactly as ours, but without any IF statements. Let's call the above program Solution #1. Here are six more solutions. In each solution, we will show only the *changes* to be made to Solution #1.

Solution #2 uses the On ... GOTO statement and the SGN function to replace the IF statements in lines 510, 520, and 530.

```
510 ON SIGN(X) + 2 GOTO 520, 530, 540
520 PRINT "YOUR NUMBER IS NEGATIVE" : GOTO 310
530 PRINT "YOUR NUMBER IS ZERO" : GOTO 310
540 PRINT "YOUR NUMBER IS POSITIVE" : GOTO 310
```

You may also delete lines 700 and 710 of Solution #1.

In Solution #3, we use ON ... GOSUB instead of ON ... GOTO.

```
510 ON SIGN(X) + 2 GOSUB 530, 540, 550
520 GOTO 310
530 PRINT "YOUR NUMBER IS NEGATIVE" : RETURN
540 PRINT "YOUR NUMBER IS ZERO" : RETURN
550 PRINT "YOUR NUMBER IS POSITIVE" : RETURN
```

Again, delete lines 700 and 710 of Solution #1.



Solutions 1 through 3 are the types of solutions we have become accustomed to seeing for this problem. We rarely see the solutions which follow.

Solution #4 uses a string array, A\$( ).

```
130 A$(1) = "NEGATIVE"
140 A$(2) = "ZERO"
150 A$(3) = "POSITIVE"

510 N = SGN(X) + 2
520 PRINT "YOUR NUMBER IS" A$(N)
```

Or, we can pack information into a single string and pick off the appropriate substring, as in Solution #5, below.

```
130 AS = "NEGATIVEZERO POSITIVE"
510 N = SIGN(X) + 1
520 PRINT "YOUR NUMBER IS " MID$(AS, 8*N+1, 8)
```

In some BASICs, of course, you will use the SEG\$ function where we have used MID\$. Or, if you have an ATARI 400 or ATARI 800, replace line 520 with the following.

```
520 PRINT "YOUR NUMBER IS" ; AS(8*N+1, 8*N+8)
```

Next, here is Solution #6

```
510 N = SGN(X) + 2
520 RESTORE
530 FOR K = 1 TO N
540 READ AS
550 NEXT K
560 DATA NEGATIVE, ZERO, POSITIVE
570 PRINT "YOUR NUMBER IS" AS
```

Hmmm... let's combine the ideas of Solutions 5 and 6. Here is Solution #7.

```
130 FOR K = 1 TO 3
140 READ AS(K)
150 NEXT K
160 DATA NEGATIVE, ZERO, POSITIVE

510 N = SIGN(X) + 2
520 PRINT "YOUR NUMBER IS" A$(N)
```

A few versions of BASIC have a GOTO exp statement, where exp may be any BASIC expression. Solution #8 uses a GOTO exp statement along with clever use of two logical operations.

```
510 "GOTO 530 + 10*(X>0) - 10*(X<0)"
520 PRINT "YOUR NUMBER IS NEGATIVE"
525 GOTO 310
530 PRINT "YOUR NUMBER IS ZERO"
535 GOTO 310
540 PRINT "YOUR NUMBER IS POSITIVE"
545 GOTO 310
```

It works like this:

VALUE OF X	VALUE OF (X>0)	VALUE OF (X<0)
Negative	0	1
Zero	0	0
Positive	1	0

So, you can now compute the corresponding values of  $530 + 10*(x > 0) - 10*(x < 0)$  in line 510.

Some people used logical operations in a different way, illustrated by Solution #9, below:

```
130 AS = "POSITIVE ZERO NEGATIVE"
510 N = 11 + 10*(X>0) - (X<0)
520 PRINT "YOUR NUMBER IS " ; AS(N, N+10)
```

Watch out! Line 510 assumes that a logical expression such as  $x > 0$  or  $x < 0$  has the value -1 when true, or 0 when false. Yes, some computers use 1 for true; others use -1 for true. Solution #9 assumes values for  $x > 0$  and  $x < 0$ , as follows:

VALUE OF X	VALUE OF (X>0)	VALUE OF (X<0)
Negative	0	0
Zero	0	-1
Positive	-1	0

Is your computer a 1 or a -1?

Oh yes, almost forgot. In line 520 you may have to change A\$(N,N+10) to MID\$(A\$, N, 10).

Well, our ingenious readers sent us some real surprises. Solution #10 is Richard Gilman's "off-the-wall" approach. He used ON-ERROR to achieve a GOTO. The appropriate errors are generated by  $A = \text{SQR}(x)$  and  $A = 1/x$ .

```
10 REM *** SOLUTION C TO PROBLEM #1
20 CLS
30 REM *** ASK FOR A NUMBER, X
50 INPUT "NUMBER, PLEASE "; X
100 REM *** CHECK FOR NEGATIVITY. LINE 120 CAUSES ERROR IF X
    IS NEGATIVE.
110 ON ERROR GOTO 500
120 A = SQR(X)
200 REM *** CHECK FOR ZERO. LINE 220 CAUSES ERROR IF X IS ZERO
210 ON ERROR GOTO 600
220 A = 1/X
300 REM *** IF NUMBER IS NEITHER NEGATIVE OR ZERO IT MUST BE POSITIVE.
310 PRINT "YOUR NUMBER IS POSITIVE"
320 GOTO 710
500 REM *** ERROR SUBROUTINE CAUSED BY NEGATIVE NUMBER IN LINE 120.
510 PRINT "YOUR NUMBER IS NEGATIVE"
520 RESUME 710
600 REM *** ERROR SUBROUTINE CAUSED BY ZERO IN LINE 220.
610 PRINT "YOUR NUMBER IS ZERO"
620 RESUME 710
700 REM *** GOTO 'ASK FOR A NUMBER, X'
710 GOTO 50
999 END
```

Solution #11. Bob Baker and students Sarah Barter and Debbie Hauth used only string functions and arithmetic to isolate the minus sign and compute an index for the string to be printed.

```
100 REM***PROBLEM #1 POSITIVE, NEGATIVE OR ZERO
110 REM***RECREATIONAL COMPUTING, JAN/FEB 1980
120 CLS
300 REM***ASK FOR NUMBER, X
310 PRINT: INPUT "NUMBER, PLEASE"; X
500 REM***TELL WHETHER NUMBER IS POSITIVE, NEGATIVE OR ZERO
510 XS = STR$(X) : REM CONVERT X TO A STRING
520 XS = LEFT$(XS,1)
530 XS = XS + "1" : REM MAKE X A POSITIVE OR NEGATIVE 1
540 Y = VAL(XS) : REM Y BECOMES + OR - DEPENDING ON X
550 X = X*(-1) : REM CHANGE THE SIGN OF X
560 XS = STR$(X) : REM CONVERT -X TO A STRING
570 XS = LEFT$(XS,1) : REM ISOLATE THE SIGN OF -X
580 XS = XS + "1" : REM MAKE -X A POSITIVE OR NEGATIVE 1
590 Z = VAL(XS) : REM Z IS + OR - DEPENDING ON -X
600 X = (Y-Z) / 2 + 2 : REM THIS EQUATION CAUSES THE ORIGINAL X
    TO BE 1, 2 OR 3. ZERO AND NEGATIVE ZERO
    MAKE Y AND Z BOTH EQUAL TO POSITIVE 1.
    THEREFORE Y - Z BECOMES 0 FOR ZERO,
    2 FOR A POSITIVE NUMBER, AND -2 FOR A
    NEGATIVE NUMBER
610 XS(3) = "YOUR NUMBER IS POSITIVE"
620 XS(2) = "YOUR NUMBER IS ZERO"
630 XS(1) = "YOUR NUMBER IS NEGATIVE"
640 PRINT XS(X)
700 REM***GOTO 'ASK FOR NUMBER, X'
710 GOTO 310
999 END
```

Solution #12. Perhaps the most exotic solution was contributed by John J. Wavrik. Since his TRS-80 (level I) doesn't have the computed GOTO X command, he wrote a GOTO with a dummy line number: GOTO 100. Then, he peeked around till he found where the digits of the dummy line number were stored and poked in a new, computed value. Here are his comments and the program.

"One of the handy features I sometimes would like in BASIC but which isn't found in my TRS-80 is a GOTO X... a 'compute GOTO' The effect of such an instruction can be



obtained by a program which actually changes itself. Here is an example of such a program for a Level II TRS-80:

```
10 INPUT N
20 S = SGN(N) + 49
30 POKE 17173,S
40 GOTO 100
100 PRINT "NEGATIVE";GOTO 10
101 PRINT "ZERO": GOTO 10
102 PRINT "POSITIVE" : GOTO 10
```

The number 17173 is the location in memory of the last '0' in line 40. When you run the program, the POKE statement on line 30 will change the GOTO to read GOTO 100 if the number you put in is negative, GOTO 101 if the number you put in is zero, GOTO 102 if the number you put in is positive."

To use this idea in your version of BASIC, you must write a program with a "dummy value" for the address on line 30. Then you must find out where in memory the BASIC text is stored and, within that block, where, in particular, the GOTO 100 command on line 40 is. I used FOR I= 17129 TO 17200: PRINT I; PEEK(I), : NEXT to get a list of memory addresses and their contents. The TRS-80 stores the BASIC text starting at 17129 for a Level II machine. The text consists of a 2 byte number showing the address of the start of the next line, a 2 byte number representing the line number, a collection of bytes for the instructions and, finally, a 0 to indicate the end of the line. All of the BASIC commands are relayed by 1 byte "tokens." 141 is the token for GOTO. Numbers (like 100) are just represented by the ASCII codes for the digits (so 49 48 48 is the representation for 100). Once the location of the final 0 is determined, the dummy address in the POKE statement is replaced by the correct address."

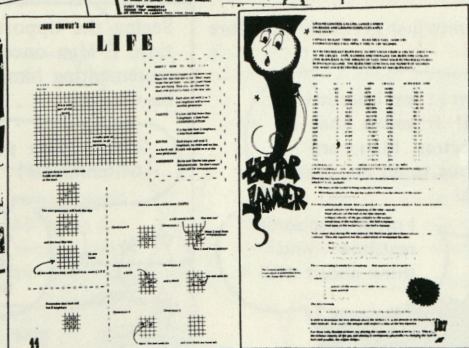
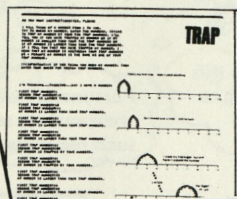
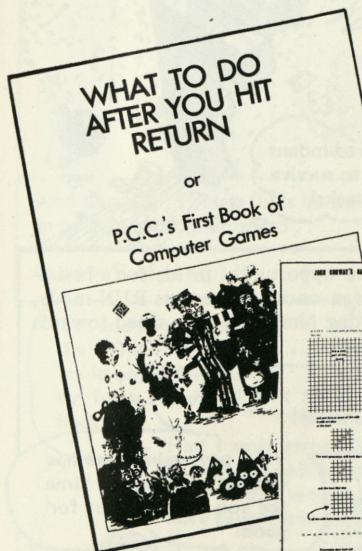
Solution #13. Raymond Schuman's two line solution was the shortest submitted. His program uses SGN(X) to compute the ASCII number for the appropriate response character '+', '0' or '-'. Exceedingly ingenious!

```
10 INPUT X
20 PRINT CHR$( - 4*SGN(X) - SGN(X) + 48)
```

Here are our problem-solvers for PROBLEM #1 showing who used which method or methods.

	1	2-3	4-5	6	7	8	9	10	11	12	13
Baker, et al.									X		
Bishop			X								
Canova								X			
Daley						X					
Day						X					
Gilman		X						X			
Hawkinson			X								
Johnson								X			
Langford			X								
Lemay									X		
Lindsay		X	X			X					
Perkins						X					
Schuman											X
Schwanbeck									X		
Wavrik		X	X							X	
The Editors	X	X	X	X	X						

## What To Do After You Hit Return



PCC's First Book of Computer Games

Fifth Printing

*What To Do After You Hit Return* is a compendium of games for kids and adults— games to develop decision making and problem solving skills for new and unexpected situations.

Includes complete listings in BASIC. 184pp. 5th printing. Retail price: \$10.95. (Add \$1.25 book rate or \$2.50 UPS for shipping. California residents add 6% sales tax.)

Please send me \_\_\_\_\_ copy(ies) of PCC's *What To Do After You Hit Return* for which I enclose \$10.95 plus shipping charge for each copy ordered.

Send your order to People's Computer Company, Dept. J7, Box E, Menlo Park, CA 94025.

Name \_\_\_\_\_

Address \_\_\_\_\_

City/State \_\_\_\_\_ Zip \_\_\_\_\_

Or Charge it! ☐ Visa ☐ MasterCard ☐ American Express

No. \_\_\_\_\_ Expires \_\_\_\_\_

Signature \_\_\_\_\_



# FORTMAN

Volume III

Episode 12

By L. SCHNEIDER & T. VOROS

In our last episode, Fortman had at last penetrated the innermost chambers of the Glitchmaster's stronghold—the Master Control room of Tesla Tower—to be confronted, at last, by the Glitchmaster himself!

And yet...even F-Man must PAUSE in surprise...for the greatest power of a Glitch lies in being able to appear as who was never expected...and certainly the ridiculously harmless-looking creature encountered by Our Hero was *not* what he was expecting!

Nor was he by any means harmless! For no sooner had F-Man stated his program objective of terminating the Glitchmaster's control of MicroprocessorLand than the latter began to produce an amazing array of weapons—each coming literally out of nowhere, and each allowing the Glitchmaster to launch yet another deadly attack against Our Hero!

Yet in the face of an Off-By-One Inducer, a Pointless Pointer, and a Noise Pulse Generator, Fortman Man remains untermiated! And yet, the Glitchmaster only announces calmly that he has only just begun...and they may now proceed with the *real* battle!

Let's see now...I wonder how you'd fare against my Data Destroyer, F-Man?

And proceed he does...with an amazing series of strange and deadly weapons, one after another...and each of them aimed quite accurately at Our Hero!

And yet, moving at top execution speed, Our Hero manages to narrowly miss each new assault, processing himself through a seemingly dimensionless array of Error Enhancers, Input Invalidators, Format Fizzlers, Logic Losers, Code Crunchers, Timing Tweakers and scores of others...

...or maybe a Dimension Distorter will do the trick

.... and here's an oldie-but-goodie: my Function Freezer!

I'll bet *this*'ll slow down your throughput rate a bit or two!

Whew! Catching this Glitch isn't going to be as easy as I thought!

Fortunately my redundant code enables me to survive such recursive attacks!

And then, with unexpected suddenness, the assaults end! And although worn down by a bit or two, F-Man again repeats the iteration of his terms...

Yet the reply comes just as calmly as before...

Seeing the opportunity in his foe's hesitation, F-Man once again enters RUN mode, incrementing himself at high speed towards his opponent...

Well then...are you ready to give up yet?

Put an end to this noisy conflict and come quietly!

Hmmmm...an interesting problem! It seems that *you*, Fortman Man, are impervious to all my best weapons!

I must take some realtime and think about this!

That's what *you* think, Glitchmaster!

Your processing time has expired...now it's time for you to be output for good!



Yet in the face of F-Man's swift attack the Glitchmaster makes no move to defend himself... nor does he make any attempt to relocate himself out of the execution path of the oncoming foe....

This is it, Glitchmaster!  
You have executed all your tricks; and now its *my* turn to....

Ah, but there you are incorrect, F-Man, You seem to forget that I always have the final, last resort of a Glitch! You see...

Wha... where'd he go? I was certain that....

Of course! I should have remembered! A glitch *always* disappears just as you get close enough to fix it for good!

Perhaps my memories were not as fully restored as I had thought!

And abruptly, F-Man finds himself standing in an empty program space....

Quickly F-Man explains....

... and so you see, the Glitchmaster could not defeat me by force... so he fled!

Had I only remembered in time his ability to escape in that way, I may have caught him yet!

Yes, F-Man, trying to catch a Glitch can be very aggravating... but it is of no matter! We shall destroy the devices here and disconnect Tesla Tower... so it will be impossible for him to ever RETURN to our land!

... I can ...

... disappear!

But the space is not empty for long, as within nanoseconds there comes a CALL from the corridors outside... and F-Man needs little time to identify its source code....

Fortran Man! Where are you?

In here, Billy!

And as the General departs to regroup his Resistance decades and take on the job of altering the current status of Capital City to handle the recently liberated Lower Cases, it is Linpa's turn to speak out....

Yes, F-Man, the threat is gone... and the Land of the Little People is once again noise-free... thank to you!

And now that *that's* done with, Billy and I have some catching up on old times to do... don't we, Billy?

Another job terminated successfully!

eh? er... what?

Thank heavens we traced you to this location! Is your code intact? What a mess!

Yes... and where is the Glitchmaster???

And soon thereafter he is linked to by not only Billy Basic, but Linea and General Wirewound as well!

So once again Fortran Man is left alone in the now-deserted program space, responding to the falling edge of the clock as it descends behind the distant Monolythic Mountains, and Peace settles into every gate and latch element in the land....

Now I can RETURN at last to my home in 360 City, and relax for a while before carrying on the fight for Truth, Justice and the Algorithmic Way! Only.... I wonder... what will come next?

And that, dear readers, only time will tell. For now, though, it is time for us to say:

STOP  
END





ALSO PUBLISHED BY PEOPLE'S COMPUTER COMPANY:

# DR. DOBB'S JOURNAL of COMPUTER Calisthenics & Orthodontia

If you're tired of plugging canned programs into your computer, *Dr. Dobb's Journal* is the magazine for you! We publish software that will allow you to get "inside" your machine. You will be able to make it do anything you want it to do. *Dr. Dobb's* is a serious magazine for those people who really want to learn as much as they can about their microcomputer.

Recent issues have included: • An All-CP/M Issue, including an overview of CP/M-compatible software, the history of CP/M, utilities, interpreters, etc. • Mathematical Typography by Donald Knuth • A Microprocessor-Assisted Graphics System • Interfacing a Home Computer to a Large Scale System • Improvements to 8080 PILOT

## OUR READERS SAY:

"I received my introduction to your remarkable journal and was as pleased as a Lovecraft wizard coming upon a copy of the *Necronomicon*. Really, yours is the only real computer journal going. *Dr. Dobb's* is to the other "small systems" periodicals as *Scientific American* is to *Popular Mechanics*."

"*Dr. Dobb's Journal* is a truly impressive periodical unlike others I've read. It contains valuable and useful software. I look forward to its delivery every month."

See insert card inside this issue.



Recreational Computing  
Box E  
Menlo Park, CA 94025

SECOND CLASS MAIL  
MAGAZINE

94062WARRED05

A

289

DR. D J WARREN

COMP

Address Correction Requested



# Recreational COMPUTING

# 6 ISSUES Just \$12

(SAVE \$3 OFF NEWSSTAND PRICE)



Name

Please Print

Address

City, State, Zip

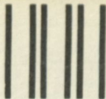
Signature

Offer good in USA only.  
Foreign Rates upon Request.

## Send No Money, Mail Today!

J9





NO POSTAGE  
NECESSARY  
IF MAILED  
IN THE  
UNITED STATES

# BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 756 MENLO PARK, CA

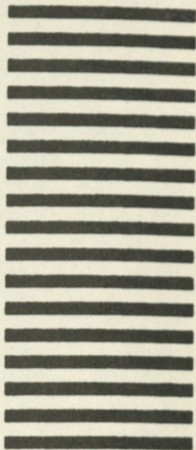
POSTAGE WILL BE PAID BY ADDRESSEE

**Recreational  
COMPUTING**



For the imaginative small computer user

SUBSCRIPTION SERVICE  
P.O. BOX E  
MENLO PARK, CA 94025





# Dr. Dobb's Journal of Computer Calisthenics & Orthodontia

## 10 ISSUES Just \$19

Name

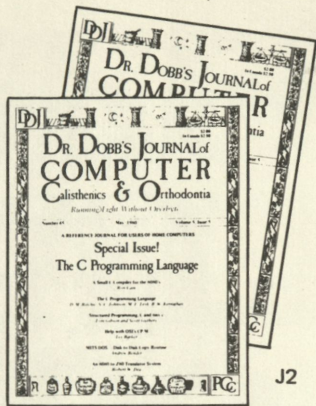
Please Print

Address

City, State, Zip

Signature

Offer good in USA only.  
Foreign Rates upon Request.



J2

## (Save \$7 off newsstand price)

## Send No Money, Mail Today!





NO POSTAGE  
NECESSARY  
IF MAILED  
IN THE  
UNITED STATES

**BUSINESS REPLY MAIL**

FIRST CLASS PERMIT NO. 756 MENLO PARK, CA

POSTAGE WILL BE PAID BY ADDRESSEE

**Dr. Dobb's Journal**  
of Computer Calisthenics & Orthodontia

**SUBSCRIPTION SERVICE**  
**P.O. BOX E**  
**MENLO PARK, CA 94025**

