



Dr. Dobb's Journal

of Computer Calisthenics & Orthodontia



Dr. Dobb's Journal is a lively forum for the more advanced home computerist.

FREE SOFTWARE

COMPLETE SYSTEMS & APPLICATIONS SOFTWARE

User documentation, internal specifications, annotated source code. In the two years of publication, *DDJ* has carried a large variety of interpreters, editors, debuggers, monitors, graphics games software, floating point routines and software design articles. Recent issues have highlighted:

- An Interactive Timeshared 8080 Operating System
- Tiny Grafix for Tiny Basic
- The Heath H-8 System
- A KIM/6502 Line Editor
- LISP for the 6800
- Dumping Northstar Disk Files
- A 1K Utilities Package for the Z80

Dr. Dobb's Journal is published 10 times a year by People's Computer Company, a non-profit educational corporation. For a one-year subscription, send \$15 to *Dr. Dobb's Journal*, Dept. 5M, 1263 El Camino Real, Box E, Menlo Park, CA 94025 or send in the postage-free card at the center of this magazine.

REVIEWS

"A publication that is a must for everyone in the hobbyist world of computers. Don't miss it."

'Newsletter'
The Digital Group

"THE software source for microcomputers. Highly recommended."

'The Data Bus'
Philadelphia Area Computer Society

"It looks as if it's going to be THE forum of public domain hobbyist software development. Rating - ☆☆☆☆"

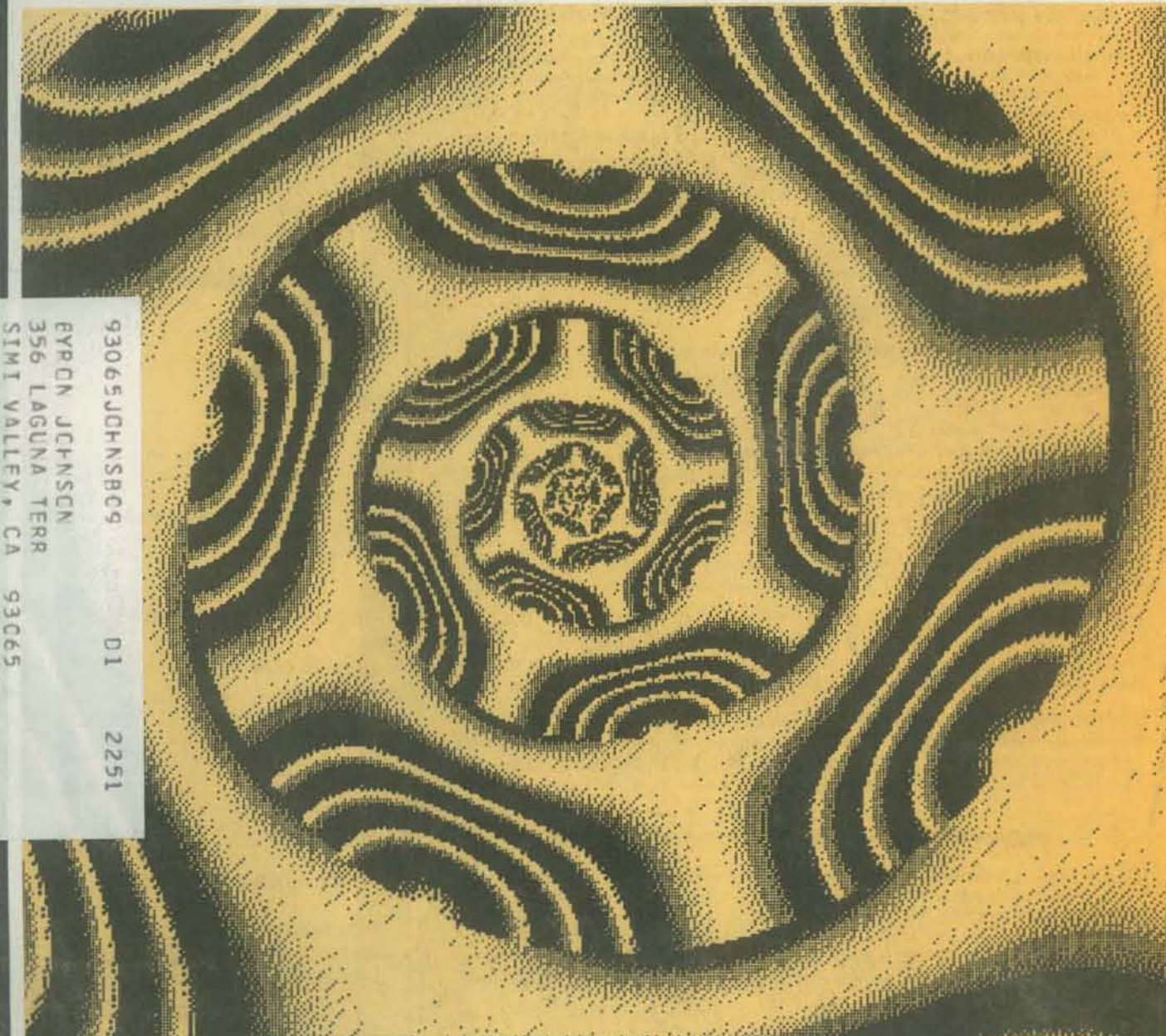
'TRACE'
Toronto Region Association of Computer Enthusiasts

"The best source for Tiny BASIC and other good things. Should be on your shelf."

'The Computer Hobbyist'
North Texas (Dallas) Newsletter

Recreational COMPUTING

FORMERLY PEOPLE'S COMPUTERS
VOL 7 NO 5 ISSUE 38 MARCH - APRIL 1979 \$2.00



93065JQHNSBC9
BYRON JCHNSCN
356 LAGUNA TERR
SIMI VALLEY, CA 93065
D1 2251

Artificial Intelligence
Calculator Comics

'Lord of the Rings'
Chess Reconsidered

SUBMITTING ITEMS FOR PUBLICATION

LABEL everything with your name, address and the *date*; tapes should also include the program name, language and system. TYPE text if at all possible, double-spaced, on 8 1/2 x 11 inch white paper. DRAWINGS should be as clear and neat as possible in black ink on white paper.

LISTINGS are hard to reproduce clearly, so please note:

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

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

SUBSCRIPTION INFORMATION

U. S. RATES

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

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

BACK ISSUES

\$2.50 each: Vol. 6, Nos. 1, 2, 3, 4, 5; Vol. 7, Nos. 1, 2
\$3.00 each: Vol. 7, Nos. 3, 4
Outside the U.S. add \$.50 per issue.

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

FOREIGN DISTRIBUTORS OF RECREATIONAL COMPUTING

Vincent Coen, LP Enterprises, 313 Kingston Road, Ilford IG11PJ, Essex, UK; Rudi Hoess, Electronic Concepts PTY Ltd., Ground Floor Cambridge House, 52-58 Clarence Street, Sydney NSW 2000; ASCII Publishing, 305 HI TORIO, 5-6-7 Minami Aoyama, Minato-Ku, Tokyo 107, JAPAN; Eastern Canada: Liz Janik, RS-232, 186 Queen St. W., Suite 232, Toronto, ON M5V 1Z1; Western Canada: Brian Wiebe, Kitronic Audio, Digital & Electronic Components, 28236 26th Ave. RR5, Aldergrove BC VOX 1A0, Canada.

Printed by Nowels, Menlo Park, CA

Recreational Computing (USPS 356870) is published bimonthly by People's Computer Company, 1263 El Camino Real, Box E, Menlo Park, CA 94025. People's Computer Company is a tax-exempt, independent, non-profit corporation, and donations are tax-deductible. Second class postage paid at Menlo Park, California, and additional entry points. Copyright © 1979 by People's Computer Company, Menlo Park, California.

STAFF

EDITORS
Bob Albrecht
Louise Burton
Ramon Zamora
ART/PRODUCTION MANAGER
Sara Werry
PRODUCTION ASSISTANT
Carole Cullenbine
ARTISTS
Jay Bonner
Aleca Harrison
Ann Miya
Judith Wasserman
TYPESETTERS
Phyllis Adams
Gavin Cullen
Renny Wiggins
PROOFREADER
Nancy Heubach
CIRCULATION MANAGER
Michael Madaj
BULK SALES MANAGER
Cynthia Kosina
SPOT EDITOR
Harry Saal
PUBLISHER
Willard J. Holden

And a special thanks to all the other folks at People's Computer Co.: Ezra Berg, Claire Connor, Delia Daniels, Loic Jassy, Ann Merchberger, Curtis Roads, Suzanne Rodriguez, John Strawn, Denise Winn.

RETAINING SUBSCRIBERS

David R. Dick
Mark Elgin
John B. Fried
Scott B. Guthery
W. A. Kelley
Frank Otsuka
Shelter Institute
Brett Wilson

SUSTAINING SUBSCRIBERS

Byte Publications
Paul, Lori and Tom Calhoun
Dick Heiser, The Computer Store

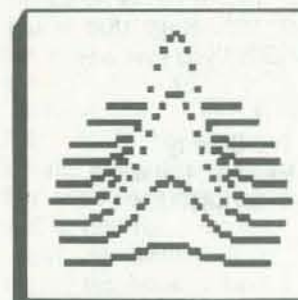
Recreational COMPUTING

Volume 7 Number 5
March-April 1979

formerly
**people's
computers**



pg. 8



pg. 21



pg. 48



pg. 16

COVER
Art by Dan Silva
Design by Ann Miya

Special Features

- 8 ARTIFICIAL INTELLIGENCE, Part 2 by David Cudhea
Can you put a computer in left field?
- 16 DATABASE by Warwick Slough
Some experiments are best left undone
- 22 BEASTIARY From SAM76 by Charles Ehrlich & Joseph Tulloch
What to do if your Translatus Uggligr goes wild
- 24 LABOR RELATIONS by Ralph Roberts
Heavy metal talks a hard line
- 32 FORTRAN MAN by Lee Schneider & Todd Voros
Our hero RETURNS!
- 34 RUNEQUEST by Steve Perrin & Friends
Rurik the Restless & Rikur the Rambler leave home
- 48 THOSE UPPITY CALCULATORS by Chris Juricich & Gil Morales
Kinky circuits recharge the comics
- 59 COLOR 'YOUR OWN' GRAPHICS by Dan Silva
Instructions on coloring the cover

Articles

- 17 RAISING CROPS FOR SPACE SETTLEMENTS by Annita Harlan
Agriculture goes to the stars
- 18 A NEW ALGORITHM FOR CHESS, Part 2 by David Chelberg & David Watters
It's easy, when you play on 20 boards
- 28 SARGON DIARY By Mike Gabrielson
Six months in the life of a busy student
- 37 EPIC GAMES ON MODEST COMPUTERS by Arthur Wells, Jr.
Will work on immodest ones, too
- 40 UNIVERSE by Les LaZar
A sure-fire cure for the Star Trek blahs
- 46 Easy POKEing with Applesoft™ BASIC by Chuck Carpenter
POKEing an Apple is as easy as PEEKing a PET

Games & Stuff

- 15 AIR RAID by Milan D. Chepko
Cut the wrong wire and the bomb goes BANG!!
- 21 TRS-80: 3D PLOTS by Milan D. Chepko
Put fancy graphics on your screen
- 27 TWO RECURSIVE FUNCTIONS IN BASIC by Dennis Allison
Yes, Martha, some BASICs are recursive
- 31 SLOT by Donald C. Hoffman
Turn your TRS-80 into a bandit
- 52 SPOT: THE SOCIETY OF PET OWNERS AND TRAINERS by Harry Saal
Pointers, products, peeves and paws
- 55 APPLE ROSE by Jim Day
A rose is an Apple is a rose is an Apple

Departments

- 4 EDITORS' NOTES AND LETTERS
- 50 DRAGONSMOKE
- 56 REVIEWS
- 60 ANNOUNCEMENTS



Editors' Notes

"Recreational *what?*" a friend exclaims when I tell him where I'm working these days. "Recreational *computing?* You mean there's actually a market for that sort of thing?!"

That doubter should see our mailbox. Every day it's bulging with letters and manuscripts from people who want to get in on the act. Strange as it may seem to the uninitiated, recreational computing is a hot property. In fact, we're getting so many ideas on the subject that the present size of the magazine seems downright claustrophobic. (Expansion will have to await the settlement of the West Coast paper strike, however.)

It wasn't easy as we approached press-time. How were we going to fit in a limited space all the things we *could* have in the March-April issue? More accurately, all the things we *wanted* to have. But we courageously attacked the puzzle: If this article goes in, then not that one—too similar. If we cut that TRS-80 game, its companion piece has to go, too. Have we got anything for kids? Are there enough programs? Can that short story hold till next time? Oh, and don't forget the comics! ... Change, cut, re-mix, look at squint-eyed, juggle, mix again, stop.

It's a process that's often frustrating, but we love it. In fact, we wish you'd make our job even tougher. The more material we have to choose from, the better the magazine will be. So this is a call for more articles, letters, programs, reviews. For suggestions, corrections, denunciations. When you get inspired by something you've discovered at your computer or irate about something you've read on computing (here or elsewhere), write us. The best articles always spring out of intensely felt situations. Take, for example, Mike Gabrielson's SARGON diary (page 28), which is full of the triumphs and setbacks all programmers know so well. Or our three opinionated movie reviews of *Lord of the Rings* (page 56). Or Les LaZar's vision of the game to end all games (page 40).

An incredible windfall this issue is our cover. That design (as well as the one to color on page 59) is the work of Dan Silva of Menlo Park. None of the editors had seen anything quite like Dan's collection of computer generated graphics when he stopped by the office one day to show us a sampling. (In a coming issue, Dan will tell you how he does them.) Once again we were faced with a terrible choice; we could only use two designs, and there were at least 25 we wanted.

This job is getting more complicated—and more fun—all the time. I guess it's the fun of PCC that still amazes me. I mean, putting together a magazine, while it has its rewards, is usually just plain hard work. But this publication is different. Which is why, on my 1040 this year, I'm listing my occupation as "recreational editing."

— Louise Burton (plus Bob Albrecht and Ramon Zamora).

Letters

GETTING HIGH ON SPACE AND COMPUTERS

Thanks so much for the good words and fine publicity on L-5 in the Nov-Dec issue of *People's Computers*. There is an impressive overlap between spacers and computerbums. You should expect our next newsletter soon.

Keep up the great job. *RC* and *Dr. Dobb's* are my two favorite personal computing magazines—potent forces for the demystification of technology that is not only high but can get you that way.

Stella Calvert
Michigan L-5 Society
Box 126 Michigan Union
Ann Arbor, MI 48109



DRAGON'S FIRE IGNITES THE NORTH COUNTRY

Upon meeting Mr. Bob Albrecht on the seventh of November and being struck with his vibrant energy, I decided that part of that energy came from working on magazines like *People's Computers*. I would therefore ask that you please send me a few (five) of your subscription forms so that I and some of the other characters here at M.S.D. can tap into this energy source. Your prompt attention to this request will be appreciated.

Benet R. Freund
Minnesota School for the Deaf
Faribault, MN 55021

Letters Letters Letters

PROGRAMMERS ANONYMOUS SEEKS FELLOW TRAVELERS

I would like to point out three *minor* mistakes in your book: *What To Do After You Hit Return*. First of all, in the listings in the back of the book, you neglected to give a listing of *HANGFL*, as you had indicated on page 19 (my school's system is Hewlett-Packard 2000 A, B, C, D, and F).

Second of all, your 'STTR1' is neither the longest nor the best *STAR TREK* simulation around. In our timesharing system we have a *STAR TREK* game that busts the britches off *STTR1*. It is... *TREK73!* It is approximately 30,000 words long, and is five programs chained together with common variables. I have enclosed a run for you; hopefully, you will think to include it in the next printing of the book. I have a punch of *TREK73*, but not of the other four it is chained to. It's great...

Third, three students in our school (including myself) helped with the development of *WUMPUS*. We desire a separate balloon on page VI which looks like this:

PROGRAMMERS ANONYMOUS
Mahwah High School
Mahwah, NJ

We, that is, Programmers Anonymous, have written a program which simulates the logged-off mode of the system, enticing "victims" to log-in, and their efforts are printed on a file. We have obtained several log-ins this way. (We have 30, including the system-operators.)

Programmers Anonymous is a "society" run solely by the students, and is open to anyone from any part of the country who sends us some reasonable proof that he is a Master Programmer like all of us in N.J. (we do have standards). Please put our membership plea in an issue of your magazine. We want to make P. A. a national organization!

Apart from those three *minor* mistakes, *What To Do...* is fantastic! Whoopee! Yippee!! I've already started putting in *Trader*.

Programmer Anonymous #8
429-16 Franklin Turnpike
Mahwah, NJ 07430

P. S. P. A. already has 36 members in various N. J. schools.

If you truly are Master Programmers, you should be able to create your own listing for HANGFL, which is nothing more than a file of words from HANGMN. Go to it!

PLAUDITS FOR THE PET

In recent issues, you have printed two letters from a minister of a non-Christian sect, each complaining in bitter terms about Commodore Business Machines and the PET computer. As a pastor and proud PET owner, I'm writing to say his attacks are unfair.

As luck would have it, my PET needed repairs twice during its 90-day warranty period. Both times the needed repairs took less than a week plus shipping time, and have been without charge other than shipping one way. I have talked with several people at Commodore, both before and after buying my PET, and found them most helpful on the telephone. The four PET owners I know personally are all quite pleased with their computers.

The PET is an incredible piece of work, as will become even more obvious after its printer and disc systems are released. For \$800, I'm glad to have a computer at all, let alone such an advanced model. For such a price, it's foolish to also expect the sort of continuing consumer service that IBM offers.

Rev. James Strasma
Grace United Methodist Church
120 W. King St.
Decatur, IL 62521

UP THE ESTABLISHMENT!

I appreciated Warner Mach's defense of the computer establishment (*People's Computers*, May-June 1978). I also am part of that, I guess, and the users I work with seldom even know what they want. That is, they are not trained in problem definition and data correlation. They are expert in their own area, but have little idea how that corresponds to other fields.

That doesn't mean I'm putting them down. I consider user education to be a primary part of my job, preferably expressed in the user's specialty. But I think I can be forgiven some pride in being familiar with many areas other than D.P.

Enough tirade; just accept my thanks for a good magazine.

Dana Taylor
3677 Radnor Road
Radnor, OH 43066

A GRIEVOUS GRAMMATICAL FAULT! (#2 IN 'THE NAPKIN LETTERS')

Dr. Robert ALBRECHT
% PETS
897 Sta. Cruz Ave.
MENLO PARK, CA 94025

My dear Dr. Albrecht:
I am greatly upset that your fine magazine would publish a letter in which there is a glaring syntactical mistake. I refer to Prof. Conlan's letter. His second sentence is in fact a sentence

fragment. I suppose this is but one more example of all those examples which gave fuel to prop. 13.

Sincerely
M. W. Perry
(a fellow drinker) *Pedagogus emeritus*

Letters Letters Letters

FIRST BUSINESS, THEN AN ADVENTURING DREAM

Several things . . .

First . . . Do you still have a bookstore? I had to go back several issues to find it, so I'm asking before ordering. [No]

Second . . . Do you or anyone out there in the readership have any programs in RPG II? [We don't. Anyone out there?]

Third . . . Get the Dragon back. [WE ARE BACK!]

Fourth . . . Would anyone out there like to sell back issues of *People's Computers* at reasonable rates? [They're also available from PCC for reasonable rates.]

Fifth . . . I would like to say that I enjoyed your articles about cheap ways (approaches) to the mechanics of robotics and would like to see more of the same.

Sixth . . . Has anyone seen the article, "Computer Poker," in the July '78 issue of *Scientific American*? Mr. Findler has done some work with intelligent programs playing poker with each other and humans in order to find out about the human thinking process, simulating it so machines can be used to make human decisions instead of humans. He has several types of players, both static and learning—all with different strategies.

The point I would like to make is that similar things could be used in an Adventure-type game, where you have a program or programs along on your quest. The programs would experience along with you and would be able to help you out (perhaps even have insight or *aha!* spurts of ideas). The programs could have human frailties and faults built in, or they could be mechanical life forms with no frailties—at the expense of compassion and love. You could even have a situation where you (the human) are fighting a war, complete with armies, etc. for any time period you would wish to program—against a program that could either attack or defend against you on a terrain of

yours or the programmer's choice. Each of you could have "program subcommanders" and so on—as far as the eye could see.

Since learning programs get larger and larger, they might pose a problem for micros. But I'll stop rambling and wait to see what the rest of you have to say.

Seventh . . . BAC, if you see this, write.

Douglas "Dit-Dit" Philips
McCombs Road
R.D. #2 Box 329
Venetia, PA 15367

P.S. Would like to hear from anyone interested in trading programs or buying or selling them. . .

CHEERS FOR FANTASY GAMES—AND SOME QUESTIONS, TOO

I've been reading your publications for about two years now and I'm keenly interested in your recently increased coverage of FRP (Fantasy/Role-Playing) games. These games, and computers, have been two of my major hobbies for years, and I'd be fascinated by any attempt to fuse them. I have a few comments on the topic:

Your publication of excerpts from *Runequest* and *Authentic Thaumaturgy* (Nov-Dec '78) did little to explain the actual workings of an FRP adventure to the neophyte. Someone with no experience in fantasy simulation gaming might still come away with the feeling that *Runequest* is sort of an updated Monopoly. I realize, however, that the task ahead of you is extremely difficult and should be done in small, easily digestible steps. I personally could not even make the attempt, as my experience shows that the best way to describe an FRP game is through an actual adventure. I put to you, however, that if you fail to interest a significant portion of your readership in gaming, the subject will die due to apathy.

I'm wondering why you chose to concentrate on *Runequest*, as opposed to the far more established Dungeons and Dragons. D&D has been around since 1972, compared to *Runequest's* six months, and the number of D&D players grows every day. *Runequest*, although unquestionably better than the original D&D, is just too new to attract many hardcore D&Ders. The average Dungeonmaster, after putting hundreds of hours (and sometimes dollars) into the creation of his fantasy world, is unlikely to drop everything and start from square zero merely because a more definitive set of rules has been released. I know I wouldn't.

I do have some ideas on computer-assisted FRP gaming, but unfortunately I have no computer facilities of my own. Maybe I'll jot them down someday. I'd certainly like to see yours.

Kenneth Shapiro
1954 Minoru
Altadena, CA 91001

Yes, it is difficult to explain FRPs with words to beginning gamers. That is why we began with Runequest and not D&D. We believe now we have a handle on how it might be done. Look for Runequest in this issue . . .
The Editors

TWO KEYS REVERSED IN REVERSE

Changing the name of your publication is a rather drastic attempt to evade a raging hunt-'n-peck typist who put in a bum program, but it's your money!

Would you please tell me what line 206 of Ramon Zamora's "Reverse" is supposed to be? I punched (not typed!) it into my TRS-80 the minute I saw it in the last issue, and that line just won't hack it!

The error must be in the $S + A(K1)$ portion of the line, since that part seems incomplete, and my guesses for correction haven't been on target yet.

Letters Letters Letters

By the way, I've been a newsstand customer of yours (Computerland) for three issues now, and I'm impressed! Hope to subscribe soon.

Robert N. Fiegel Jr.
442 Simpson
Bergstrom AFB, Texas 78743

I too hunt-'n-peck at amazing speeds (at least 10 characters/sec.) and am prone to forgetting to use the shift key. Hence, the error in line 206. The line should be $S = A(K1)$ etc. ("Reverse," PC, Nov-Dec, 1978). I am sure this will be the last missteak we well evur maked.—RZ

SHINE ON, COMPUTERS!

Any readers wishing to correspond on microcomputers and solar/alternative energy technology are invited to write me at Box 3437, Chico, CA 95927, or call me at (916) 345-8078.

Mark Miller

ANOTHER HUZZA FOR EPICS AND DRAGONS

Dear Dragons:

Everybody else in the little computer business seems to think that computers are for business and that little computers are for little businesses. We think *Recreational Computing* is right in emphasizing recreation and education with little computers, although there does seem to be a lot of material directed at the education of children by others, compared with material to help one educate himself on his computer, regardless of age.

For example, the article by Dwyer and Critchfield on "distance" in error correcting was very good, but the glue thieves were more distracting than helpful (both articles in *PC*, Nov-Dec 1978). On the other hand, we have been enjoying the articles on epic gaming immensely (. . . hard to do immense epic games on

little computers, though . . .), and we are looking forward to the Dragon's exposition of the Dungeonmaster via computer. We don't care how many heads the dragon has. Like a famous English author who wrote epic tales, we desire dragons.

Cecilia Ziemer Watson
Allen Watson
430 Lakeview Way
Redwood City, CA 94062

MORE ON MUSEUMS AND COMPUTERS AND KIDS . . .

I read with interest Bob Kahn's article on computers and science museums in *Computer Magazine* (April 1977) as well as the more recent version of it in *People's Computers* (Sept-Oct 1978). I am in the process of developing an economic education exhibit in which I hope to use a heuristic approach.

In doing research for both the economics (we are working with the Indiana Council for Economic Education at Purdue University) and the computer end of this exhibit, I have run across the names, descriptions and even the listings of some excellent economic-oriented simulation programs. Kingdom, by Todd Voros and Lee Schneider, was described in *People's Computers* (Sept-Oct 1978), and while its main purpose may be recreational it does involve the player in many real economic situations and with economic concepts. I understand that Star Trader also is concerned with (among other things) the economics of intergalactic trade. I would like to learn more about Star Trader as well as to locate other sources for economic simulation games of this type.

Michael P. O'Lear, Curator
Economic Education Project
The Children's Museum
Indianapolis, IN 46208

Star Trader is in What to Do After You Hit Return, to be reprinted soon by PCC. Available from us for \$10.95 a copy.

HOORAY FOR HIPPIE DROP-OUTS!

After reading the Norman Gold letter (Nov-Dec '78), I had to comment. I've been a subscriber to *DDJ* since #11, but have only gotten *PC's* from the stand. I happen to be a hippie dropout who has tried and/or lived each and every one of those cliches. I also had the same kind of subscription snafu with *DDJ* and with *IEEE's Computer*.

I'm very much in favor of people using computers as tools, and have followed the lit on PILOT and other text-using languages (even to the extent of investigating LOGLAN), and hope to do some serious work in this area when my systems are again running. Said systems are 1) an LSI-11 with dual floppies and 2) an AM-100 with dual floppies and five megabytes of hard disk. I'm aware of PCNET and plan to be part of it ASAP. The biggest holdup now is financing.

Keep IT up, lay eggs, and all that good (bleep),

Bob Williams
Illegitimi Non Carborundum
Box 69
Hollywood, CA 90028
A Division of Micromouse Enterprises

CORRECTION

The Concept Game
(Jan./Feb. 1979 issue)

SCORING RULES

- (1) Each valid line that is identified—1 Point.
- (2) If only two lines exist each worth—2 Points.
- (3) If only one line exists it is worth—3 Points.
- (4) First player to correctly claim that no valid line exists scores—5 Points.
- (5) Incorrect claim in (4) above, reduce player's score by—5 Points.
- (6) 11 Points wins the game!



ARTIFICIAL INTELLIGENCE

BY DAVID CUDHEA

PART 2

This article is the second of a two-part serialized presentation on two schools of thought in the field of Artificial Intelligence. In the last issue the basic methods of approach of two men, Edward Feigenbaum and John McCarthy, both of Stanford, were discussed. This part deals with how these men work within their particular methods and some of their basic ideas concerning how they apply their concepts.

The last article concluded with a statement of how these two men represent "polar extremities" in this complex field. Their differences generally relate to their ideas of "knowledge" and how knowledge is to be represented. This concluding article begins with McCarthy's views on knowledge.

Upon publication of the first half of this article, The Stanford Magazine's managing editor, Debby Fife, informed us that David Cudhea, the author, had passed away in July, 1978. In her letter to us, she said several things about Mr. Cudhea and his writing that were part of the reason we were originally attracted by his article. She wrote:

"He was a very fine writer with a special talent for capturing some of the most

difficult subjects, such as Artificial Intelligence and DNA research, in language which anyone could understand and enjoy."

You will no doubt agree with her as you read this half of Mr. Cudhea's article on Artificial Intelligence. — RZ

McCarthy is trying to find ways to formalize general knowledge in such a way that a computer can understand it and act upon it. "It has proved easier," he observes, "to get outstanding performances from computer programs on narrow problems than to get something that reaches the level of an average educable mentally retarded person in, say, general conversation."

Based on an article in *The Stanford Magazine*, Spring/Summer, 1978. Reprinted by permission. Some material used in this article excerpted from Margaret A. Boden, *Artificial Intelligence and Natural Man*. New York: Basic Books, Inc. 1977. Also published by Harvester Press in England in paperback (Open University Set Book) and hardcover.

Experts can be simulated but the common man cannot?

Just so, and that goes not only for general conversation but also for general activity.

"We simply don't yet know how to represent well enough the kind of facts that you might call common-sense physics," McCarthy suggests. Sitting next to his terminal, he studies me, and his eye lights on my coffee cup. "People know what to expect if some coffee is spilled," he says. "They have a rough idea of how far it's likely to spread, that it won't go shooting 20 feet across the room, and so forth. They know what to expect, and they don't have to use, say, differential equations to know it."

Aha, say I. It's like the left fielder for, say, the Boston Red Sox. At the crack of the bat — "it's a long fly ball" — he's off and running. He *knows* where it will come down.

"Well, you're right that *he's* not using differential equations," replies McCarthy. "But in fact that particular performance is something we could achieve with machines. Using the mathematics of a fly ball might work out quite well. A television camera observes the ball

leaving the bat and the computer quickly determines where it's going. So directing a robot fielder to the spot seems like it would not be difficult—*providing* the field was nice and smooth and the 'fielder' didn't have to recognize any obstacles along the way." (I picture the robot left fielder and center fielder converging. They collide and ricochet. The ball drops between them and rolls to the fence. Meanwhile robot Yankee "Reggie Jackson" rounds second base. No! He's collided with the bag and is stopped in his tracks . . .)

McCarthy's computer terminal emits a faint whirring and whistling. He turns to it, taps the keyboard for a while, and the display alters. He turns back to me.

"Representing common-sense information," he repeats, and then launches on a new tack. "Another kind of information we've been studying is knowledge about knowledge."

Suppose I ask you, as McCarthy does me: Is President Carter standing up or sitting down at this very moment? You don't know? Well, think harder, I say. And you tell me, with some exasperation perhaps, that thinking isn't going to help. "Knowing what it doesn't know is something a computer program will have to be able to do, in order to decide what then to do—perhaps look for the information it needs." (I picture the computer as Rodin's *Thinker*, reasoning on forever, and never getting an answer.) "We really don't know how to formalize that kind of thing yet."

To give you an idea of what formalizing looks like, here is one of McCarthy's recent formulations:

$$\Phi(P) \wedge \Phi(Q) \wedge (VP)Q \Rightarrow (\Phi(P) \wedge \Phi(Q) \Rightarrow \Phi(Q)) \supset (VP)(\text{knows}(\text{Mike}, P) \supset \Phi(P)).$$

Don't worry about the details. What it means is actually rather simple: All that

Mike knows is a consequence of (proposition) P and (proposition) Q. It comes from a recent paper in which McCarthy spent some time trying to formalize a mental process we all are rather good, or bad, at—namely, jumping to conclusions. The Stanford AI theorist tags this "mode of reasoning" with the label *circumscription*. Here is how he gets into it:

There is an intuition that not all human reasoning can be translated into deduction in some formal system of mathematical logic, and therefore mathematical logic should be rejected as a formalism for expressing what a robot should know about the world. The intuition in itself doesn't carry a convincing idea of what is lacking and how it might be supplied.

Does the intuition hold up? Yes, on the point that "reasoning" isn't all deduction. No, on the point that therefore mathematical logic won't be any good for robots.

INTELLIGENT MACHINE

How John McCarthy answers the question: On what basis might a machine be considered "intelligent?"

"One way is to give a . . . blackbox definition. In this case we have to say that a machine is intelligent if it solves certain classes of problems requiring intelligence in humans, or survives in an intellectually demanding environment.

"Instead, we shall use in our definition certain structures . . . such as knowledge of facts . . . [Constructing machines that manipulate facts is] the best bet both for constructing artificial intelligence and understanding natural intelligence . . .

"We shall, therefore, be interested in an intelligent entity that is equipped with a representation or model of the world . . .

"On this basis we shall say that an entity is intelligent if it has an adequate model of the world (including the intellectual world of mathematics, understanding of its own goals, and other mental processes), if it is clever enough to answer a wide variety of questions on the basis of this model, if it can get additional information from the external world when required, and can perform such tasks in the external world as its goals demand and its physical abilities permit.

"[Accordingly] . . . intelligence has two parts . . . The epistemological part is the representation of the world in such a form that the solution of problems follows from the facts expressed in the representation. The heuristic part is the mechanism that on the basis of the information solves the problem and decides what to do."

We can confirm part of the intuition by describing a previously unformalized mode of reasoning called circumscription, which we can show does not correspond to deduction in a mathematical system. The conclusions it yields are just conjectures and sometimes even introduce inconsistency. We will argue that humans often use circumscription, and robots must too. The second part of the intuition—the rejection of mathematical logic—is not confirmed; the new mode of reasoning is best understood and used within a mathematical logical framework and coordinates well with mathematical logical deduction. We think circumscription accounts for some of the successes and some of the errors of human reasoning.

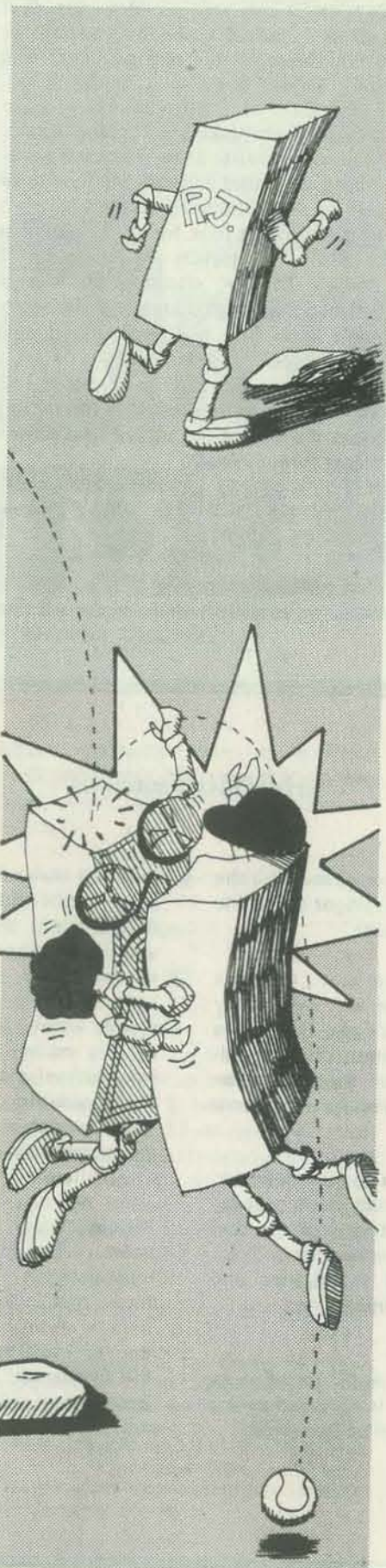
What, then, is circumscription? McCarthy describes it in plain words as follows:

The intuitive idea of circumscription is as follows: We know some objects in a given class and we have some ways of generating more. We jump to the conclusion that this gives all the objects in the class.

"Thus we circumscribe the class to the objects we know how to generate," McCarthy concludes. All that Mike knows is a consequence of P and Q, and there you have it: an example of a "circumscription schema dealing with knowledge."

Circumscription is McCarthy's baby right now. Let's move on to something related but broader, a conceptual approach that's coming on strong throughout the field of artificial intelligence. It is termed *framing*. M.I.T.'s Minsky defines it as the description—for computers—of patterns into which situations may fit. Shorthand, it's a body of context. Remember your studies of elementary French or Japanese or Urdu—all that background pertinent to in-the-restaurant, or on-the-subway, or feeding-the-yaks? Each set up a rudimentary frame. Call it an "associative scenario" if you prefer. Framed information will be easier for a computer to retrieve, and will be more like human recall, the AI researchers contend.

McCarthy, while he's not uninterested in Minsky-style framing, is more interested in something he earlier intro-



duced as "the frame problem." In the real world, we need to know which features of a context are changed by an event, which are not, and which of either may be ignored. He likes to make reference here to the classic puzzler involving missionaries and cannibals.

You remember it. Three missionaries and three cannibals want to cross a river. There's a boat available that holds up to two people. There's only one catch: The missionaries in any one place must never be outnumbered by the cannibals, for if they are, the cannibals will have dinner. (As you know, or will readily discover if you've never before tried this "brainteaser," the best first move is not to send off one missionary and one cannibal together. And several of the people here are going to make more than one trip.)

McCarthy is not concerned so much with how the puzzle is solved; he is more concerned with what information is (and is not) provided. He elaborates the "qualification problem":

In the statement of the problem, nothing is said about how boats are used to cross rivers, so obviously this information must come from common knowledge, and a computer program capable of solving the problem from an English description or from a translation of this description into logic must have the requisite common knowledge. The simplest statement about the use of boats says something like, "If a boat is at one point on the shore of a body of water, and a set of things enter the boat, and the boat is propelled to another point on shore, and the things exit the boat, then they will be at the second point on the shore." However, this statement is too rigid to be true, because anyone will admit that if the boat is a rowboat and has a leak or no oars, the action may not achieve its intended result. One might try amending the common knowledge statement about boats, but this encounters difficulties when a critic demands a qualification that the vertical exhaust stack of a diesel boat must not be struck square by a cow turd dropped by a passing hawk or some other event that no one has previously thought of. We need to be able to say that the boat can be used as a vehicle for crossing a body of water unless something prevents it. However, since we are not willing to delimit in advance possible

circumstances that may prevent the use of the boat, there is still a problem of proving or at least conjecturing that nothing prevents the use of the boat. [The] method of reasoning called circumscription, . . . is a candidate for solving the qualification problem.

The issue, the detail, and the style are vintage McCarthy. The exercise is vintage 1978 "computer science."

McCarthy and other Stanford people don't play around much anymore with building "game programs" like checkers

(Arthur Samuel, adjunct professor emeritus of computer science at Stanford, built a champion some years ago when he was with IBM), *Go*, or chess. But in a way, the chess program exemplifies the AI problem-solving problem. Even the best of chess programs, like Chess 4.6,

STANFORD'S ARTIFICIAL INTELLIGENCE PRINCIPALS



1. **John McCarthy**, professor of computer science and director of Stanford's Artificial Intelligence Laboratory. Principal designer of the LISP programming language, which is the most widely used language in artificial intelligence. Originated the concept of computer time-sharing. Published the first paper in the field of mathematical theory of computation. Clarified the separation of artificial intelligence into epistemological and heuristic components.

2. **Edward Feigenbaum**, professor of computer science and chairman of that department at Stanford. Directs the research of the Heuristic Programming Project that has pioneered in the development of concepts, methods, and programs for emulating scientific and medical reasoning processes. Prime mover in the emergence of expert knowledge-intensive problem-solving agents. His group's work on problems in chemistry, molecular genetics, crystallography, signal interpretation, and some medical specialties has made the Heuristic Programming Project the leading center for the application of artificial intelligence. With **Joshua Lederberg**, professor of genetics, and **Carl Djerassi**, professor of chemistry, he oversaw the development of the DENDRAL programs, which aid organic chemists in the determination of molecular structure of unknown compounds. They also initiated and manage the development of an NIH national research resource (SUMEX), located at Stanford, for the application of artificial intelligence to medicine and biology.

3. **Raj Reddy**, MS '64, PhD '66, professor of computer science at Carnegie-Mellon University. Developed the first automatic speech recognition system that works with connected speech over a vocabulary of a few hundred words.

4. **Thomas Binford**, research associate in computer science at Stanford. Heads a group doing research in computer vision and robotics. Systems have been developed for locating three-dimensional objects using television camera information and for assembling mechanical devices using computer-controlled mechanical arms.

5. **Bruce Buchanan**, Gr. '63, adjunct professor of computer science at Stanford. Developed META-DENDRAL, which is a learning program that writes new rules for DENDRAL to use in interpreting experimental data collected on organic chemical compounds. It learns these rules by looking at a set of data collected on known compounds and searching a space of possible rules under the guidance of a model of what chemists find acceptable. With **Ted Shortliffe**, PhD '75, MD '76, developed the MYCIN program, which is designed to aid physicians in the selection of antibiotic therapy for patients with acute infections. It is

organized around a large set of inference rules written by medical experts.

6. **Zohar Manna**, research associate in computer science at Stanford, and **David Luckham**, senior research associate in computer science at Stanford. Working on theoretical and practical methods for formally proving correctness of computer programs. This work is aimed at replacing our current imperfect "debugging" techniques with rigorous program verification.

7. **Terry Winograd**, assistant professor of computer science at Stanford. Is developing and testing schemes for making computers understand natural languages such as English. Main goals are to facilitate man-machine communication and to test theories of language understanding.

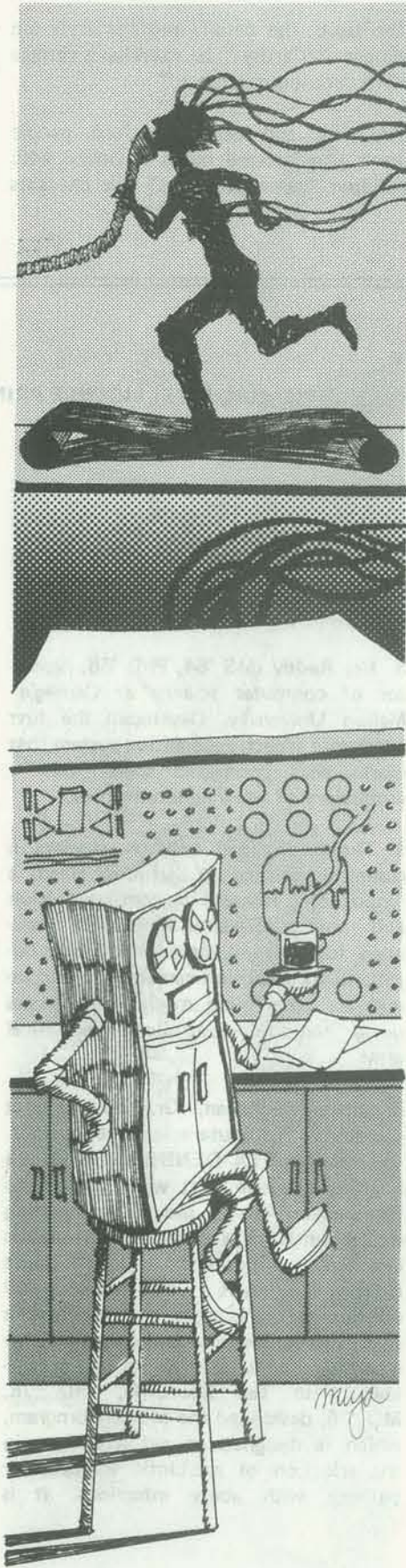
8. **Cordell Green**, assistant professor of computer science at Stanford. Heads a group that is developing an interactive program called PSI that assists a programmer in specifying what a given program is to accomplish and does much of the detailed design of data structures and the programming.

9. **Douglas Lenat**, PhD '77, assistant professor of computer science at Carnegie-Mellon University. Developed the AM program, the first system that emulated some acts of mathematical concept discovery. The program uses rules of plausibility and interestingness to search for interesting mathematical conjectures arising out of elementary set theory.

as mentioned earlier, are better at tactics—protecting a piece, attacking a piece—than at planning long-range strategy and sensing subtle advantages. In *Artificial Intelligence and Natural Man*, Margaret Boden sums up this way:

Chess-playing programs... do not use mere brute force methods. They set bounds to the lookahead (which may be relaxed in certain circumstances). They use evaluation functions embodying knowledge about what moves are likely to contribute to king safety, material balance, center control, and so on. They take account of their opponent's likely actions instead of merely the legality of moves open to them. They can sometimes exploit their opponent's weaknesses in much the same way as does the checker player, learning to attack individuals in an aptly idiosyncratic manner. Some of them can benefit from expert advice. And some of them play very respectable chess... However... there are far too many alternatives to be considered for the program confidently to pick the right one; and as it cuts down the range of alternatives, it is as likely as not to lose the best move, so throwing out the baby with the bathwater. A grandmaster was asked how many moves he considered before choosing, and his answer was: "One—the right one!" H.J. Berliner has pointed out that master chess players develop global perceptual schemata in terms of which they can see threats and opportunities on the board much as a lesser mortal can see a complex emotional response to a cartoon face. Until chess masters have made more progress in articulating the knowledge they use, there is no question of its being represented computationally.

Can, in fact, experts in various real-world disciplines be helped to make their knowledge explicit? The evidence from Stanford unequivocally demonstrates that the answer is "yes." How do you go about it, and then program a computer to use it? Ed Feigenbaum explains: "Experience has shown us that [expert] knowledge is largely heuristic: experiential, uncertain, mostly 'good guesses' and 'good practice' in lieu of facts and rigor... What the masters really know is not covered in the textbooks of the masters. But we have learned that this private knowledge can be uncovered by the careful, painstaking analysis of a second party."



For example, look at the recent work of Doug Lenat, now of Carnegie-Mellon, when he was taking his PhD at Stanford. Lenat devised a mathematical discovery program and dubbed it AM: It conjectures "interesting" concepts in elementary math. AM doesn't prove theorems; it conjectures interesting theorems to prove. In other words, it tries to capture the expertise of a creative professional mathematician at work. It does rather well at it. Given some concepts of elementary set theory, for example, it conjectured number theory: "add," "multiply," "primes," and the unique factorization theorem. It conjectured, and revived interest in, the interestingness of a concept similar to primes called "maximally divisible numbers," previously studied only by a talented but obscure Indian mathematician, and somewhat forgotten.

Too esoteric? Try this one, then. Feigenbaum's knowledge engineers are currently applying their techniques to the highly specialized knowledge of a respiratory disease physician at the Pacific Medical Center in San Francisco. Together the AI specialists and the respiratory specialist, Dr. Robert Fallat, are creating a system called PUFF. Though it's in a rather preliminary stage, it looks promising. To begin with, the patient breathes for a while into a spirometer, the data from which is used to calculate the flow rates and volume of air in his or her lungs. Various other measurements are made and the data, along with items from the patient's history—sex, age, number of years of smoking, etc.—are fed to the computer.

PUFF's task is to make the diagnosis: What kind of pulmonary function disorder, if any, does the patient have? PUFF will answer, in English, and provide the usual kind of summary of interpretation that a referring physician would expect.

PUFF operates by virtue of rules. They take the logical form IF... THEN... Here is an example:

IF:

- 1) The severity of obstructive airways disease of the patient is greater than or equal to mild, and
- 2) The degree of diffusion defect of the patient is greater than or equal to mild, and

WHY DIDN'T YOU TREAT THE VIRUS? ITEM-1 is Virus associated with meningitis. I know of no antibiotics which are effective for treating meningitis caused by Virus.

DID YOU USE THE PATIENT'S AGE? RULE 315 used: the age of Fred Jones. This information also would have been used in: 545, 544, 535, 534, 518, 517, 327, 326, 317, 316, 314, 265, 256, 247, 245, 150, 135, 125, 102, 66, 55, 54. However, none of these succeeded in the context of Fred Jones. If you would like an explanation for why any of these rules failed, please enter their numbers.

545.
RULE 545 would be applied to: CUL-

TURE-3. Rule 545 was applied to CULTURE-3, but failed due to clause 6 (the infection was acquired while the patient was hospitalized).

HOW DID YOU DECIDE TO TREAT MYCOBACTERIUM-TB? ITEM-2 is Mycobacterium-tb associated with meningitis. I decided to treat ITEM-2 on the basis of clinical evidence only...

HOW DO YOU TREAT PNEUMOCOCCAL MENINGITIS? For treatment of DIPLOCOCCUS-PNEUMONIAE in MENINGITIS the drugs of choice (ranked by relative efficacy, assuming sensitivity) are:

1st choice: PENICILLIN
AMPICILLIN

2nd choice: NAFCILLIN
CARBENICILLIN
3rd choice: CHLORAMPHENICOL
ERYTHROMYCIN
CEPHALOTHIN
TETRACYCLINE

Duration of therapy: Treat for at least 5 afebrile days or for a minimum of 10-14 days.

DOES THE AGE OF THE PATIENT EVER TELL YOU ANYTHING ABOUT THE IDENTITY OF THE INFECTING ORGANISM?

The following rules use: the age of the patient to conclude about: the identity of the organism—327, 326. Which of the rules do you wish to see?
ALL.

3) The tlc (body box) observed/predicted of the patient is greater than or equal to 110 and
4) The observed/predicted difference in rv/tlc of the patient is greater than or equal to 10
THEN:

- 1) There is strongly suggestive evidence (.9) that the subtype of obstructive airways disease is emphysema, and
- 2) It is definite (1.0) that "OAD, Diffusion Defect, elevated TLC, and elevated RV together indicate emphysema," is one of the findings.

"Everything PUFF knows about pulmonary function diagnosis is contained in 55 such rules," Feigenbaum notes. "No textbook of medicine currently records these rules. They constitute the partly public, partly private knowledge of an expert... and were extracted and polished by project engineers.

One hundred cases, carefully chosen to span the variety of disease states with sufficient exemplary information for each, were used to extract the 55 rules. As the knowledge emerged, it was represented in rule form, added to the system, and tested by running additional cases.

The expert was sometimes surprised, sometimes frustrated, by the occasional gaps and inconsistencies in the knowledge, and the incorrect diagnoses that were logical consequences of the existing rule set. The interplay between knowledge engineer and expert gradually expanded the set of rules to remove most of these problems.

Technically, PUFF, like the MYCIN system from which it borrows, is a "backward-chaining inference system." As Feigenbaum explains, it seeks a line of reasoning that is valid, based on its rules and on the patient data supplied to it. Perhaps not surprisingly, such programs help—or force—the experts to clarify their theories of medicine or other "domains." PUFF will be even more helpful at that and at its primary function—diagnosis as an aid to a physician—when it is programmed to explain why it comes to the conclusion it reaches.

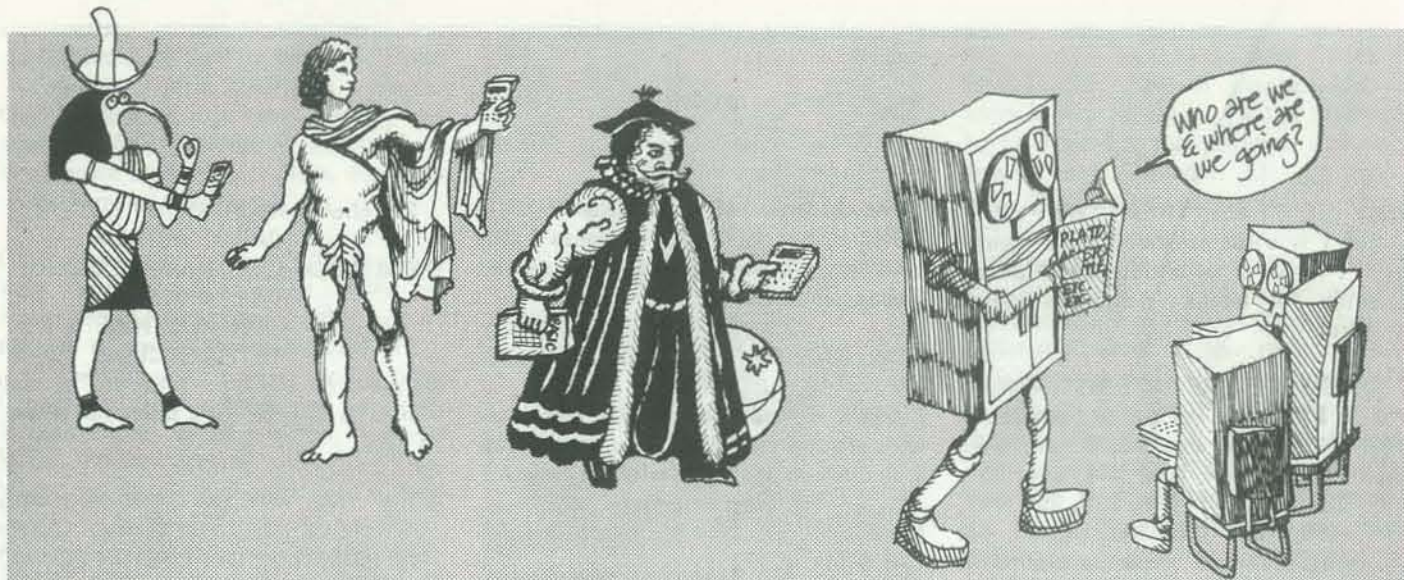
MYCIN can already do that and the feature is important (see box). As Margaret Boden points out:

The physician asks MYCIN for advice on the identification of microorganisms

and the prescription of antibiotic drugs, and also for explanations of its advice expressed at the appropriate level of detail.

MYCIN's explanatory capability is not put in merely for show. It enables the physician rationally to reject the program's advice if she disagrees with specific aspects of its clinical rationale. It helps nonspecialist doctors to learn more about the complexities of diagnosis and therapy in this class of diseases. And it allows human consultants to make general improvements in the program, by telling it about relevant knowledge that they realize in specific cases to be missing or inadequately stated. For example, it can be pointed out to MYCIN that such-and-such a drug should not be given to patients with a particular allergy or rare heart condition... A program like MYCIN, then, is much more than a computerized table of bugs-and-drugs, and can be written only with the help of artificial intelligence techniques...

For Feigenbaum, and for many other workers in the field of AI, "intelligence" necessarily embraces intelligibility. His definition, phrased some years ago, is



as follows: "Intelligent action is an act or decision that is goal-oriented, arrived at by an understandable chain of symbolic analysis and reasoning steps, and is one in which knowledge of the world informs and guides the reasoning."

There we are again: *knowledge of the world*. Feigenbaum may concentrate on "domain-specific" matters, McCarthy may concentrate on "domain-general" matters (my own loose phrase), but both are concerned principally with how knowledge can be acquired by, represented in, and used appropriately by computer programs (and humans) to reason and to take action. A colleague once told Feigenbaum he should call his work "epistemological engineering," which Feigenbaum, as noted, simplified to "knowledge engineering." John McCarthy has described his work as "applied epistemology." Epistemology: the study of the nature and grounds of knowledge. What do we know? How much do we know? How do we know what we know? Ask all these questions again, now, and for "we" substitute "computer programs."

Artificial intelligence specialists, in short, now come to the forefront of a long, long, graybeard line of thinkers extending back over thousands of years. McCarthy puts it best: "Designing computer programs capable of acting intelligently in the world requires commitments about what knowledge is and how it is obtained. Thus some of the major traditional problems of philosophy arise in artificial intelligence."

It remains to be seen, of course, whether or not those who call themselves artificial intelligencers will have more success than those who have viewed the problem as exclusively human (or for that matter, at root, divine). However... here again is Boden:

The "tin can" computer buffs predict that by the end of the century (or 2001, if you prefer) a single silicon chip computer, only a few millimetres square, will be able to follow 20 million instructions a second using its 65K (65,000 cells) of internal memory store—and it will sell for about one U.S. dollar. (Programs currently considered "large" use only about 250K... [and] there will probably be several 1,000K programs in existence by 1980.) And research on programming languages continues apace... ambitious tongues that make [current ones] look like childish lispings. Whether or not it ever achieves the highest hopes of its enthusiasts, and whether or not it will soon be able to emulate HAL, artificial intelligence therefore can be expected to become more "natural" in character than it is at present.

Machines, in sum, are going to be "thinking" more and more and better and better. Does the prospect please you?

"The prospect of having machines as smart as men triggers some worries," McCarthy acknowledges. "One fear is that things may get out of control, that someone may use robots to conquer

the world... A second fear is not concrete but rather more abstract—people may not like to believe that it's even possible for machines to do what humans can do, whether or not they actually do it."

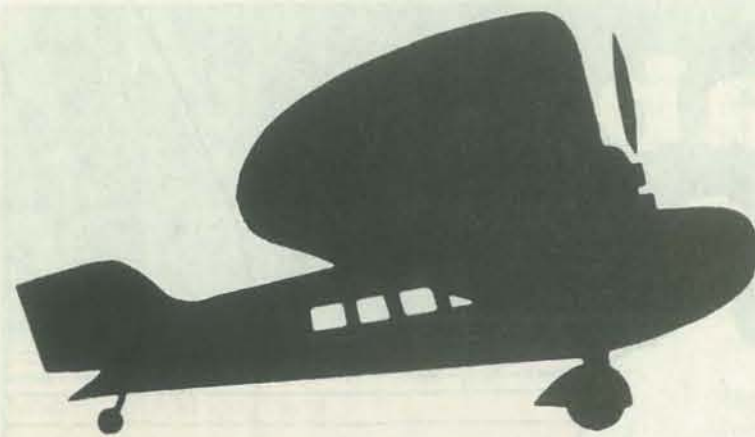
McCarthy treats the first case this way: "As true high-level artificial intelligence approaches, and we understand it better, then we can *decide* how to get the benefits and protect ourselves from dangers."

The second case? McCarty is blunt. "As to the philosophical unpleasantness felt by some about the fact that the world happens to be such that a machine can think: that's tough."

We'll just have to face this fact?

"Sure," says McCarthy. "Just as strong men have had to face the fact that machines are even stronger."

Tough is the mind of McCarthy. Tenderer, in many ways, is the mind of me. I neither wish to be, nor expect to be, replaced by a computer program. But the argument can and should be made that just as machines, for all the faults of "the technological society," have in fact enhanced and extended humankind's physical abilities in beneficial ways, so also may machines supplement our own intellectual abilities. Can and will "intelligent agents" be of help to us? The answer being shaped at Stanford is a firm and definitive "yes." □



AIR RAID!

BY MILAN D. CHEPKO

Remember the old movie where a team of specialists defuses bombs in London? Some bombs are duds; some are live and get defused; some blow up.

Milan takes you to London, 1943. You get the chance to defuse an unexploded bomb left by a recent AIR RAID. Anyone with a TRS-80 sound generator can make the program really go BANG!!

Dr. Chepko is a medical doctor living in Minnesota and a frequent contributor to RC. This program is written in Level II BASIC and there is information at the end of the program (lines 2010-2050) for converting to Level I. -RZ □

Variables

- A() = status of each wire (0=cut)
- C = number of wires cut
- D = DUD (if=0)
- T = Trip wire
- W = next wire to be cut

```

2 'AIR RAID, BY MILAN D. CHEPKO
3 '119 BELLEVILLE CRT. THIEF RIVER FALLS, MN 56701
10 DEFINT B-Z:CLS:PRINTTAB<15>"LONDON, 1943":PRINT
11 PRINT"WHILE MAKING YOUR INSPECTION TOUR DURING AN AIR RAID
12 PRINT"YOU DISCOVER AN UNEXPLODED BOMB. NORMALLY, YOU WOULD
13 PRINT"LEAVE IT ALONE, BUT YOU CAN HEAR IT TICKING, AND
14 PRINT"HEADQUARTERS INFORMS YOU BY RADIO THAT THE BOMB
15 PRINT"DISPOSAL SQUAD CAN'T GET THERE FOR AT LEAST AN HOUR.
16 PRINT"PRINT"LIFTING OFF THE INSPECTION COVER, YOU CAN SEE 5
17 PRINT"WIRES AND THE TIMER, SHOWING ONLY 60 SECONDS LEFT. THE
18 PRINT"EXPERT (SAFE AT HEADQUARTERS!) ADDS THE FOLLOWING:
19 PRINT:INPUT"<HIT 'ENTER' TO CONTINUE>";A#
20 RANDOM:C=0:CLS:FORI=1TO5:A(I)=1:NEXTI
21 T=RND<5>;D=RND<4>;IF D=2 THEN D=0
22 PRINT"1";TAB<16>:"YOU MUST CUT 3 OF THE WIRES
23 PRINTTAB<16>"TO DEFUSE THE BOMB. ":PRINT"2"
24 PRINTTAB<16>"THERE IS ONE 'TRIP WIRE' THAT
25 PRINT"3";TAB<16>"EXPLODES THE BOMB IF CUT. ":PRINT
26 PRINT"4";TAB<16>"CHEER UP... SOME BOMBS ARE DUDS!!
27 PRINT:PRINT"5"
28 FOR X=3TO25:FOR Y=1TO26 STEP6:SET(X,Y):NEXTY:NEXTX
29 PRINT@640,"WHICH WIRE WILL YOU CUT?":INPUTW
30 IF W>5 OR A(W)=0 THEN GOTO200
31 A(W)=0:RESTORE:FOR I=1TO W:READ Y:NEXTI
32 FOR I=1TO13:RESET(I,Y):NEXTI
33 IF W=T GOTO300
34 C=C+1:IF C<3 GOTO200
35 PRINT@640,"CONGRATULATIONS!!! THE BOMB IS DEFUSED!":GOTO500
36 IFD>0 CLS:PRINTCHR*(23):PRINT@470,"BANG!!":GOTO500
37 PRINT@640,"YOU WERE LUCKY!!! YOU CUT THE TRIP WIRE,
38 PRINT" BUT THE BOMB WAS A DUD!!!
39 FORX=1TO2000:NEXTX:GOTO100
4000 DATA 1,7,13,19,25
4100 '
4200 '
4300 '
4400 'WRITTEN IN LEVEL II, BUT CAN BE USED IN LEVEL I BY:
4500 ' DELETING "DEFINT B-Z" FROM LINE 10
4600 ' DELETING "RANDOM" FROM LINE 100
4700 ' DELETING "PRINT CHR*(23)" FROM LINE 300
4800 ' REWRITING LINE 210 USING "< >+< >" FORMAT
4900 ' CHANGING "@" TO "AT" IN PRINT STATEMENTS

```

Prints instructions.

Determines if "DUD."

Sets random trip wire.

Prints final instructions.

Prints wires.

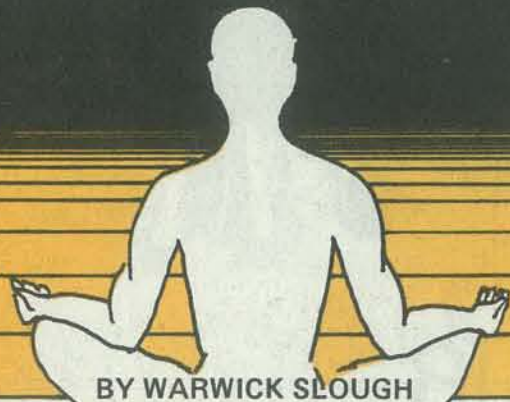
Requests input.

Decodes input, cuts the wire selected.

Prints results.

Level II to Level I conversion instructions.

database



"... a mind forever voyaging through strange seas of thought alone..." Wordsworth.

Have you ever been in a computer center late at night? Remember the quiet hums, clicks and noises of the machinery? Remember the sense of presence in the room with you? Your thoughts about the machine as it went about its tasks? Your sense of wonder and fear at the possibilities and capabilities of that electro-mechanical creature?

Mr. Slough, of Twickenham, Middlesex, England, returns in this issue with a story that you might say brings all those late night experiences to life. It is an inventive journey through a truly unique "sea of thought."—RZ

On Wednesday morning arrangements had been completed for the Minister to see work going on in the computing area. After a general rundown which came with insipid brown tea, the multimicroprocessor project had been mentioned. "But what is a multimicroprocessor?" queried the Minister.

"Each processing unit has its own data store and programs to enable any correlations within this store to be made, in addition to being coupled to all other units," enthused Joe. "We've fed in the most comprehensive data available on psychology, physiology, life-chemistry, nuclear physics, astronomy and palaeontology. Now we're ready to examine the multiprocessor angle—and maybe something new will turn up."

"I see," said the Minister. (Which meant that he didn't.)

As the days passed, ever more complex relationships were established from programmes written to exploit the multitudinous data stores. At the end of three months, a discussion was taking place in Joe's office. Alan Merryweather, a fair-haired gangling youth was summarising. "As far as our analyses go, we have, on the basis of our chosen data, a species in which development has been so rapid that a basic conflict between the old and the new must occur. Furthermore, violence and aggression are endemic to the model."

"But surely the parallel processing capability available here is far greater than that of the human brain?" interjected Joe.

"Certainly," replied Merryweather with some acerbity, "but the data stores are still common."

There it was—the most comprehensive correlations of existing knowledge indicating that evolutionary processes had not eliminated aggression and violence. For a stable model to emerge, some other data must obviously be considered. But what? Joe pondered. "What we need is a data bank representing human feeling," he expostulated to Alan the next day. "Oh, I know pain and misery are covered in the statistical stores giving deaths, genocide and war casualties, but somehow we have to input to the system personal feelings of pain and anguish. We have to make our microprocessors feel pain."

"And how, pray, is that to be accomplished?" demanded Merryweather.

"Well there's a guy I know at Berkshire University who's got a mock-up emulating the pain centres in the human brain. If we could link that data into our multimicroprocessor system, it might give us a database closer to a human condition."

Accordingly, in the course of time, a complex arrangement was interfaced with the existing setup. Joe and Alan were excitedly arranging preliminary experiments. "If we increase this potential," Joe said, pointing with a stubby finger, "it's roughly similar to inducing an extreme anxiety neurosis in a human brain." It took another year of programming to evaluate the effect of the new database on the predictions.

For a species with a processing capacity equivalent to that constructed, evolutionary progress could now lead (over rather a long time period) to an advanced intelligent consciousness also capable of *experiencing* pain.

On an Autumn evening when swirling mists and Halloween aroused strange primordial memories, Joe was attempting to summarise the conclusions reached from the multimicroprocessor project in a report. As he sat at a desk in a corner of the laboratory, a preliminary clicking from part of the system caught his attention. Power was always connected to retain the memory—some of which was volatile. A speech synthesizer unit had recently been added, and now a low muttering of a female voice was discernible. The choice of female voice production had been Alan's. With a shock Joe realised that the complex processing unit was dreaming or at least associating images outside of the stored programs. "A pulsating universe, existing between white hole singularities," murmured the microprocessors.

Joe pondered the situation. If free association of ideas were possible, could the computer be said to possess an unconscious intelligence? If such an intelligence existed, what would be the moral consequences of destroying it? Could it be destroyed?

The following morning as he strolled between the bronze and gold splashed trees that surrounded the laboratory, he determined to test the idea. He used his personal key to enter the deserted building. Somewhere, in the distance, church bells were emphasizing the quiet of Sunday morning. In the computer room the bright October sunshine slanted through venetian blinds. A small bank of solar cells, with secondary cell storage, provided backup power to the multimicroprocessor system. Joe disconnected the main supply and waited for the power to fade.

He became gradually aware of a shadowy figure building up in a corner of the laboratory. As it intensified he identified a very old woman with wispy hair and crinkled features. From the voice unit a low murmur came, "I represent the whole of human experience of pain and anguish." Joe gazed at the image, noticed the bent despondent figure and the tears running down the withered cheeks. "But worst of all," whispered the figure, "is the loneliness."

Yes that would be the most unbearable burden of all for conscious awareness—the realisation that each and every personality is, in the last resort, *alone*.

The figure slowly lightened in intensity and then disappeared. Joe remained seated, wondering how to describe solitude of the mind in an official report. □



Raising Crops for

What to eat and how to raise it during years of space habitation are issues being addressed by a team at the University of Arizona's Environmental Research Laboratory in Tucson. Under the direction of Research Associate John Phillips, a diverse group including library specialists, biologists, and a student of psychology are considering the literature on closed environment agriculture (CEA) in search of likely food sources for space pioneers and permanent inhabitants.

The study came into existence in order to cover gaps in knowledge uncovered at the 1977 NASA-Ames Space Settlement Study which was attended by two of the team members. It has centered on collection of published works which detail the methods used in growing crops and raising animals under enclosed conditions.



U. of Arizona Space Garden

Increased yields are usually realized by CEA, and economies can be made in materials, such as nutrients and water. Recycling in controlled loops, an imperative for space habitations, naturally derives from CEA practices.

Space Settlements

With compilations of candidate species, their yields, needed foodstuffs and spatial requirements in hand, the Arizona team will be making specific recommendations to NASA in several areas.

Among them will be: utilization of space shuttles to study systems and components; development of a dedicated long-duration exposure facility for life sciences research; initiation of quantitative studies in earth-based phytotrons; continuation of interdisciplinary forums to promote developmental interaction among industrial, governmental and university personnel interested in CEA; and expansion of the data base SIRS (Space Information Retrieval System) begun during the study. □

—Annita Harlan, *L-5 News*, copyright © 1978.

a New Algorithm for Chess

BY DAVID CHELBERG AND DAVID WATTERS

PART II: A CONSISTENT DATA STRUCTURE

Here is the second installment of what promises to be an ongoing set of articles from the two Davids. As David C. turned in this article, he indicated that when they complete the discussion of the program they developed, they may continue to write on other chess topics. So look for more articles by D & D (I can't help it!) in the coming year.

With David W. at Northwestern University, perhaps we will see a few articles on Chess 5.0 and beyond. David C. is currently attending classes at Stanford University. He seems destined to write the definitive comparative analysis of the double cheeseburger offerings at the Oasis and the Dutch Goose (two infamous local beer and burger places here in Menlo Park). The main obstacle to his research project is not having a car. Mom! Dad! Are you listening?—RZ

In our last article, we discussed the history of chess programming efforts to date. We included such famous innovations as introduced by Shannon, Turing, Newell-Shaw-Simon, Northwestern University and the University of Southern California. We concluded that all of these methods were incompatible with the

computer system at our disposal (HP 2000 Access Basic System). Our purpose was to design a chess program capable of playing a challenging game of chess. We wanted a program that abided by all rules, including the time limit of two hours for the first 40 moves. With these limitations, a need for an efficient data structure became evident. This article deals with that data structure.

BEGIN WITH THE BOARD

When considering a data structure for chess, the first thing that comes to mind is how to represent the present board position. There are several ways this might be accomplished. One way is to keep a list of each piece's position and value. Another is to keep a list of the moves made in the game to the present point and have the starting position as a constant which could then be brought up to the present by merely making the moves in order. The last possibility is to represent the board as a human might see it as a two-dimensional 8×8 matrix.

The second way is unacceptable since it requires a great deal of computation to reconstruct the board position each time. Thus, the first and last methods remain. Our choice was the last method.

It seemed to us that we should try to pattern the data structure as close as possible to the way a human looks at chess. Thus, in our program, the board is represented as an 8×8 array of numeric values: 1 for a pawn, 2 for a knight, 3 for a bishop, 4 for a rook, 5 for a queen, and 6 for king. The computer's pieces are represented by negative values and the human's are positive. A zero is assigned to blank board positions. This representation, however, is not completely sufficient. The computer must also understand the relative weights of the pieces. This is accomplished by a constant matrix which assigns the following weights to the pieces: 1 for a pawn, 3 for a knight, 3.1 for a bishop, 5 for a rook, 10 for a queen, and 20 for a king.

MARKING MOVES

Moving a piece is of primary importance in a chess game. In deciding on a particular move, many possible moves must be considered. Therefore, the need arises for a comprehensive data structure to represent a particular move.

With the board represented by an 8×8 matrix, it is logical to represent a move by the matrix position of a piece. Throughout the program, a move is represented as a four-digit number. The 10^3 and 10^2

digits are the matrix coordinates of the piece's initial position, and the 10^1 and 10^0 digits are the coordinates of the position to which the piece is moving. It is preferable to use one four-digit number rather than four one-digit numbers because of the space limitation. Each new variable takes up additional space.

Now that the internal move structure is established, an external move structure needs to be developed. Two means of input exist: Standard Chess Notation and our matrix form. Any move entered in Standard Chess Notation is converted to the matrix format and all computer moves are converted to Standard Chess Notation for the final move output. Having the external format in such a manner enhances the appearance of the program and facilitates usage.

With the simple move structure defined, there still remain several problems. Namely, problems of special moves. How is castling, capturing, en passant, or pawn promotion represented?

A form exists in Standard Chess Notation that represents castling: "0-0" king side and "0-0-0" queen side, but how should this be converted to our four-digit number? We decided to also represent castling as a power of ten: 10^3 = human king's side castling; 10^2 = human queen's side castling; 10^1 = king's side castling for the computer; and 10^0 = queen's side castling for the computer. This distinguishes between castling on either side as well as either color. Since evaluating a castling move was a detailed operation, a simple way of delineating between each type is needed. This method proves most useful.

Capturing en passant is quite a different problem. It is not a normal capture, and the usual test of a piece on the final square would not work. We decided to mark any capturing en passant move by an "EP" following the input. Capturing en passant is a legal move. However, the computer is not able to identify that move by itself. Thus, to add "EP" after the input is easy for the user to remember, and computer decoding is rather simple.

Pawn promotion is a rather minor section. A normal pawn move input is sufficient, followed by a later request for input of the nature of the new piece. The value of the pawn is changed to the value of

that new piece. The computer is capable of promoting pawns. However, it automatically replaces the pawn by a queen. This completes the move structure, a vital section in any chess program.

WHAT TO FILE

The peculiarities of a specific computer system often enter into the analysis when considering data structure. In the case of our chess program, we had to decide which of two systems of storage we would use for different data. One possible storage method on our system is to use disc files. Another is to use matrices. The relative speeds and space needed for each of these storage methods must be taken into account before determining which to use.

Files on our system are organized into records of fixed size. The records are randomly accessible but the records themselves are sequentially organized. Files require some amount of core storage as a buffer. This buffer space is used during the execution of a program in accessing a file. Moving data from files to the buffer areas causes the use of files to be somewhat slower than programmatically created data structures, such as matrices. However, files are the most efficient means (when considering space used) to store any large amount of permanent data.

Matrices, on the other hand, provide quick, economical access to great quantities of data. Matrices are randomly accessible since any element may be individually retrieved. They provide a means of storing and manipulating data that must be accessed often.

Thus, files are best for long-term, mass storage, while matrices are best for storage of program-generated data that must be accessed often within a program. The main uses for files on our program are the storage of the moves made during the game, a book of openings, the state of a game, and continued games.

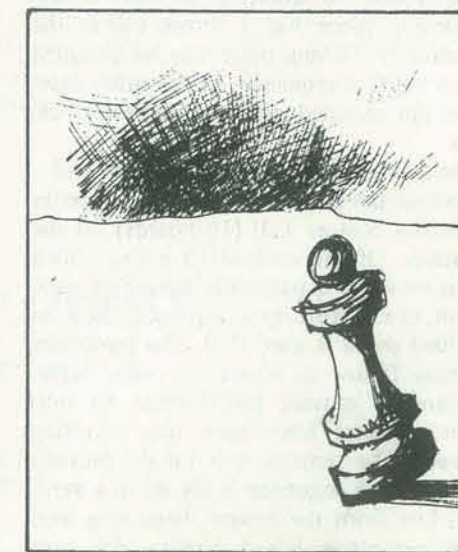
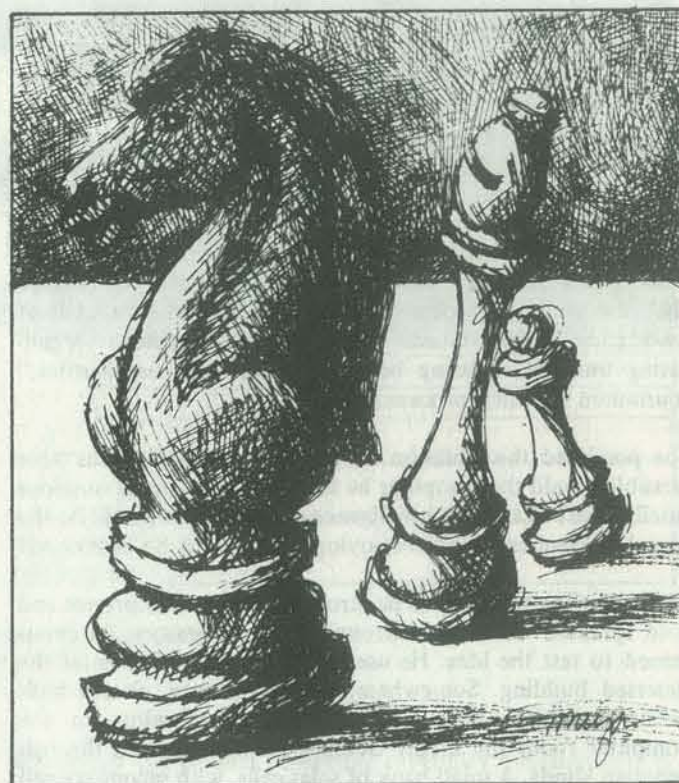
Certain files, which list the moves of the game, are necessary as a record of the game played, and for en passant captures. These files are organized so that one file (BLACK) contains the computer's moves and the other file (WHITE) contains the human player's moves. The format used to store moves is a four-digit number corresponding to a matrix move.

Another file (ZERO 2) is used to store a book of openings. The openings are stored as a set of moves. If the moves match up to a certain point, then a choice is randomly made between all possible continuations that are found in the book. The moves are stored sequentially in this file to conserve space. Organizing by record would enable quicker execution but would require five times the amount of space to store the same information.

A companion file (FINULT) to the large opening file (ZERO 2) stores the starting positions of the openings that are still under consideration. This allows quicker access to current move information. This same file has two other uses.

There are cases where the opening program is not in effect: i.e., the progress of the game exceeds the capacity of files. One such case occurs when a partially completed game is recalled. In this case FINULT is used to store the location of the game that was recalled in the continued game file (ZERO 1). This permits the player to re-store the game at the end of the session with only the new moves being added to the old game.

The final use of this file occurs when a new starting position is created for the computer. (This feature may be used to test particular sections of the program.) In this case, the file holds this new starting board position. By using this one file in a variety of mutually exclusive cases, much file space is saved. Other files are used to store chess rules, in case the human needs to be reminded, and to store the special instructions for our chess



program, i.e., how to store and retrieve continued games, create new board positions, and so forth.

MATRICES AND LISTS

Although files are used extensively throughout the program, the use of matrices is not insignificant. Our primary matrices are vital to the game strategy, in particular, the middle and end game. The most significant matrix, fundamental to our entire strategy, is called the A-matrix (Figure 1). The A-matrix has dimensions 8×160 . However, we like to visualize this as 20 chess boards stacked on top of one another, or having the dimensions $8 \times 8 \times 20$. (Three-dimensional arrays are undefinable on the HP 2000 Access System.) Pieces that can move to a particular square are stored in a vertical line in the A-matrix, corresponding to that square on the real board.

Direct moves are the most common moves considered. These include all moves of pieces that immediately protect or attack a particular square. Diagonal pawn moves are the only pawn moves desired, as this matrix is used in evaluating capture sequences.

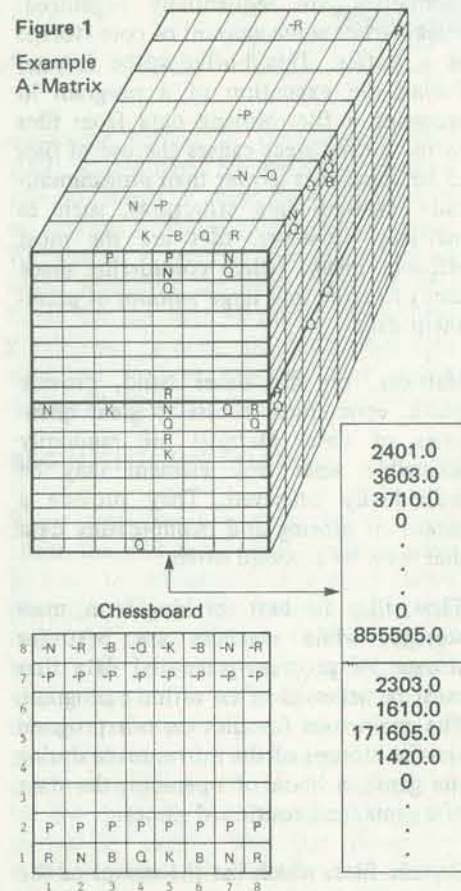
Indirect moves occur often. However, they are not as often applicable to the situation as direct moves. These include moves through one piece; i.e., a rook may be said to be indirectly attacking a certain square even though its path is blocked by a piece. These types of moves break down into two categories, whether the blocking piece has a direct move to the square in question or not. If the blocking piece has a direct move, the indirectly moving piece can be included in a capture sequence. In the other case, it is not included in the capture sequence.

The A-matrix is divided into two halves—the top half (10 boards) for the computer and the bottom half (10 boards) for the human. Of the computer's pieces which can move to a particular square in question, in a capture-type sequence, the least valued piece is used first. The remaining pieces follow in increasing order value. Indirectly moving pieces must fit into the sequence some time after blocking pieces. The same is true for the player's piece. Each sequence is placed in a vertical line from the square, beginning with the respective board number for each

side. The indirect moves that do not fit into a capture sequence are placed in the A-matrix starting from the bottom of the proper half-matrix, proceeding upwards. This provides a barrier of zeros between the two types of data for simplification of cut-offs.

Direct moves are represented by the following format: 1203.0 would mean that a piece at the coordinates (1, 2) could move to the square in question and that this piece is a knight (weight = 3.0). Indirect moves are represented: 121405.0. This would mean that a piece, a rook, (weight = 5), at (1, 2) could move to the desired square except for the impeding piece at (1, 4).

The A-matrix is used to evaluate a capture; to determine the safety of any particular position, and to evaluate the consequences of a move which would alter a capture sequence.



Two other matrices, C and D, are columnar matrices each having dimensions 2×100 . The C-matrix contains a list of the computer's moves. In the adjacent column, the tabulated value assigned to a particular move is stored. The D-matrix is similar to the C-matrix except that

the D-matrix contains the moves for the player. Moves are recorded as four digit numbers popularly referred to as the *matrix format*.

We utilize a list, Z, with dimensions 1×100 to store what we call vital protectors and discovered moves. A vital protector is a piece, which upon moving, places in jeopardy another piece that it protected prior to the move. Each move is stored as a six-digit number beginning at the top of the list and proceeding down. The location of the piece to be placed in jeopardy is stored in the 10^5 and 10^4 positions and the move is placed in its normal form. The second move-type (discovered moves) includes discovered attacks, pins and hurdles. If moving a piece in a certain manner frees a piece to attack, or places the piece behind it in danger, the move is stored in a similar format in this list, starting from the bottom and working up. This section utilizes the format of indirect moves found in the A-matrix.

We find it necessary to use two other *attack matrices*, G and H. The G-matrix stores all the possible attacking moves for the human in the form: the move X 100 plus the locations of the attacked piece. Another column is used to evaluate these moves so that a decision can be reached as to the computer's best defensive measures. The H-matrix, similarly, compiles the computer's attacking moves. However, the values assigned correspond to the C-matrix and are given to individual first moves.

NOT QUITE THE END

In conclusion, we have found that the data structure briefly described in this article has led to an efficient chess-playing computer program. The board position makes the program more readable and leads to the natural development of the A-matrix. Through the proper separation of files and matrices, the time factor involved in deciding on a move is greatly reduced. The data structure is an extremely important facet in maintaining the efficient operation of any chess program.

In the next article, we will delve into the mysterious realm of how this data structure coalesces into a logical functioning monstrosity known as our chess program. □

TRS-80: 3D PLOTS

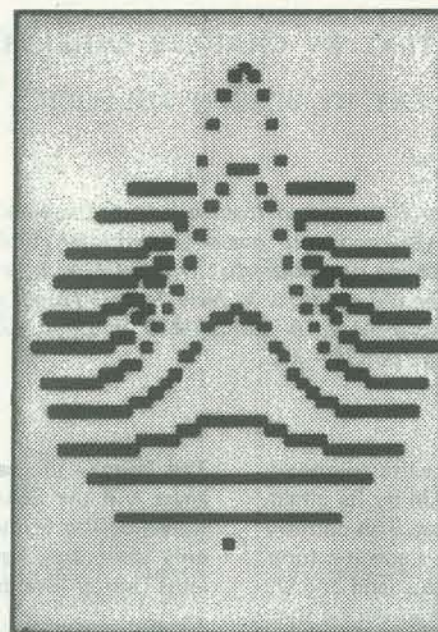
BY MILAN D. CHEPKO

This program by Milan re-creates a routine that produces 3-D plots from a set of specified functions. Dr. Chepko is one of our regular authors of TRS-80 software. His range of interests and programs seem to cover all areas: applications, graphics, games, utilities.

Why does he get so many TRS-80 articles in the magazine? Because he sends articles to us for publication!! Who else, out there, is doing TRS-80 stuff? We can't hear you... -RZ

Many interesting and useful programs have been written during the last decade. Most programs, more than a year or two old, were probably prepared for systems considerably larger than the micros we now use. Consequently, many outstanding programs languish in journals or files, unused because of the difficulties in converting them for the typical hobby system.

A good example is a program originally written by Mark Bramhall and published in *101 Basic Computer Games*. This fascinating program generates data plots that have a 3-dimensional appearance. In its original form, the program prints out asterisks in a teletype mode, taking over 30 lines for each plotted function. While this seems impressive, the plot can be difficult to follow as it scrolls by on a 16-line TV! My initial modification allowed 16 lines to print out at a time, pausing while I ran off copies on the TRS-80 Screen Printer. I then cut the sections, matched the pieces up on a sheet of paper, and taped them down. Although this operation does the job, it seems to defeat the idea that machines should work and people should think.



Finally, I realized that the TRS-80 is capable of 48 lines of simultaneous display if the graphic mode is used. I rewrote a few lines of the program to accomplish this. (See the program listing.) Now, an entire plot can be seen on the screen at one time.

Several interesting plot functions are included, and can be selectively called up by changing line 145. When you are done with a plot, simply hit BREAK to exit the program. The plot can be centered on the screen by adding the appropriate offset to Z in line 180. Also, you can even have the program automatically print the plot by adding:

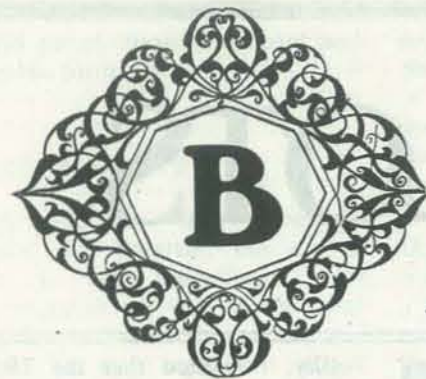
240 OUT 254, 255

In summary, we can often modify existing software to our advantage, giving new life to older programs. □

```

1 ORIGINAL PLOTTING PROGRAM BY MARK BRAMHALL
2 REWRITTEN FOR TRS-80 LEVEL II BY MILAN D. CHEPKO
3 THIEF RIVER FALLS, MN 10 DEC 78
10 CLS:DEFINT I,S
20 PRINT"THIS PROGRAM WILL GRAPHICALLY DISPLAY A 3-D PLOT OF
30 PRINT"A MATHEMATICAL FUNCTION FROM THE FOLLOWING TABLE...
40 FOR I=1TO2000:NEXT I:CLS
50 PRINT"LINE #145 SELECTS THE FUNCTION FROM THE LIST, AND
60 PRINT"NEW ONES CAN BE ADDED. TO START PLOT, TYPE 'RUN 100'
70 PRINT:LIST 300-
100 CLS:S=1
110 FORX=-30TO30STEP1.5
120 L=0
130 Y1=5*INT(SQR(900-X*X)/5)
140 FOR Y=Y1 TO -Y1 STEP-5:Z=SQR(X*X+Y*Y)
145 GOSUB 800
150 Z=INT(25+Z-.7*Y)
160 IFZ<L THEN 190
170 L=Z
180 SET(Z,S)
190 NEXT Y
210 S=S+1:NEXT X
250 GOTO250
300 '*** THESE ARE THE CURRENT FUNCTIONS ***
400 Z=30*EXP(-Z*Z/100):RETURN
500 Z=30-30*SIN(Z/18):RETURN
600 Z=30*EXP(-COS(Z/16))-30:RETURN
700 Z=30*SIN(Z/10):RETURN
800 Z=45*EXP(-Z*Z/100):RETURN
900 Z=30*(COS(Z/16))^2:RETURN
1000 Z=SQR(900.01-Z*Z)*.9-2:RETURN
1100 Z=.001*(Z*Z*Z+Z-25):RETURN
1200 Z=30*SIN(Z/5):RETURN

```



BEASTIARY

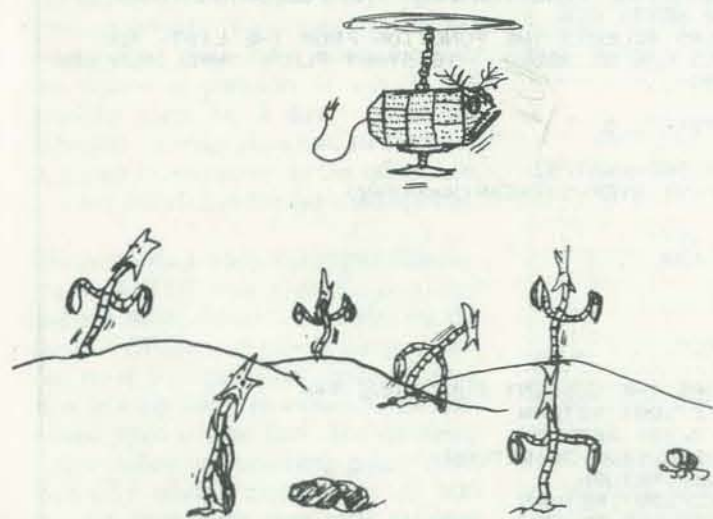
From Sam76 Book Of Beasts

TEXT BY CHARLES EHRLICH

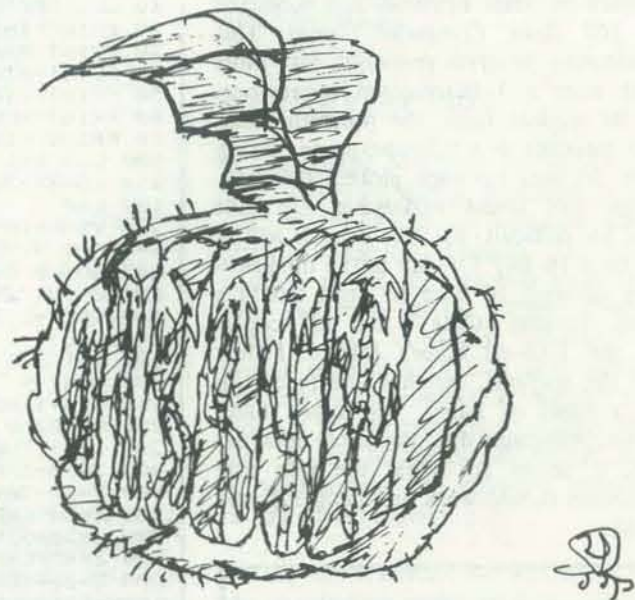
ART BY JOSEPH TULLOCH

The Beastiary is excerpted from a language manual, SAM76, created for you and your personal computer. SAM76 is an interpretive string processor that was designed by a group of 10 to 18 year old students in New Jersey. To get a copy of the SAM76 manual write to: SAM76 Inc., Box 257 RR1, Pennington, NJ 08534. -RZ

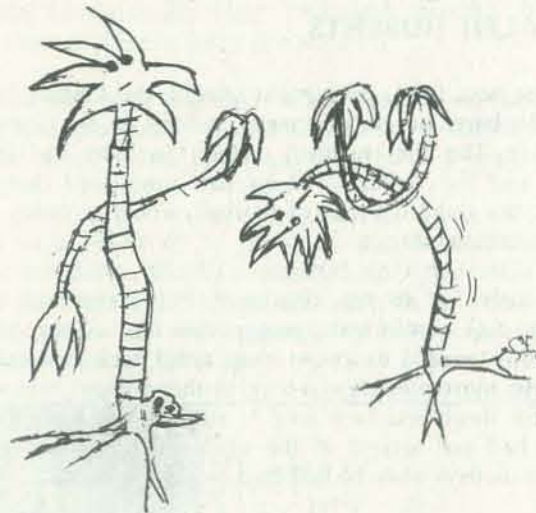
The Land of SAM. This is a high-flying satellite reconnaissance photograph of a distant land (called SAM by its inhabitants) with the residents in their natural habitat. This country is populated by crystalloids* who each have a specific function. Several people have just returned from this land and are now recuperating from terrible wounds incurred during their visit. Inside is a description of each beast's duties and their names given by the authors in both the vernacular and scientific. One cannot appreciate fully the horrible monsters the authors saw during their sojourn because all sketches were made on the run, but if you let your imagination go, you may begin to have as bad nightmares as we did and still do.



The Palma— (1. palm of the hand). The Palma (text area) beast and the Stringere (string beasts) have a symbiotic relationship. The Stringere will reside in the stomach of the Palma only after they have been prepared for storage by the Translatus Uggliqr (eval beast) for his later request.



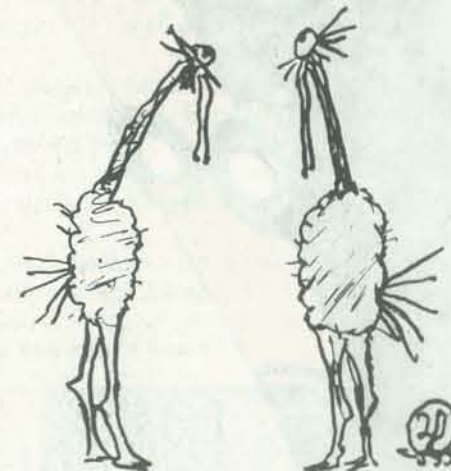
The Stringere — (1. to bind). The Stringere (string beasts), a newly discovered family of nonvertebrae, are composed of anywhere from one to many cells. These beasts are quite abundant, but are only useful in particular cases where certain strains of cells are combined. Since these nonvertebrae are so simplistic it is truthful to assume that they are insignificant and depend upon all other beasts for their livelihood.



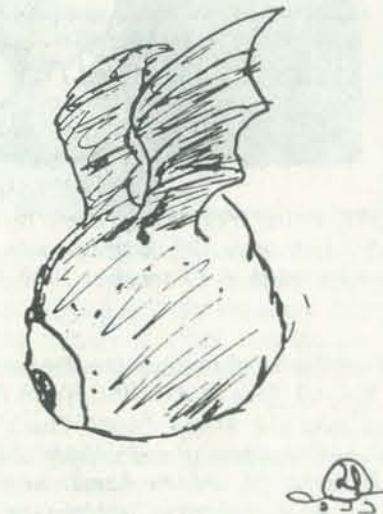
The Translatus Uggliqr — (1. translate On. Dreadful). Translatus Uggliqr arranges the pecking order. He is the sole authority on which beast grazes first. After he has devoured an active Stringere he summons by telepathic code the other beasts needed to help digest the Stringere and then excretes a simplified form of his meal, thereby distributing the seeds for another string beast.



The Lugen — (G. look). The Lugen (scanner) beast, a much misunderstood animal, gets its name from the action it performs for Translatus Uggliqr. He points out the edible Stringere to the Translatus Uggliqr for his consumption, avoiding the tasteless type.* The Lugen must select the most active edible Stringere first so Translatus Uggliqr devours them in the proper order. If the order is not followed Translatus Uggliqr goes on a rampage and becomes drastically ill.



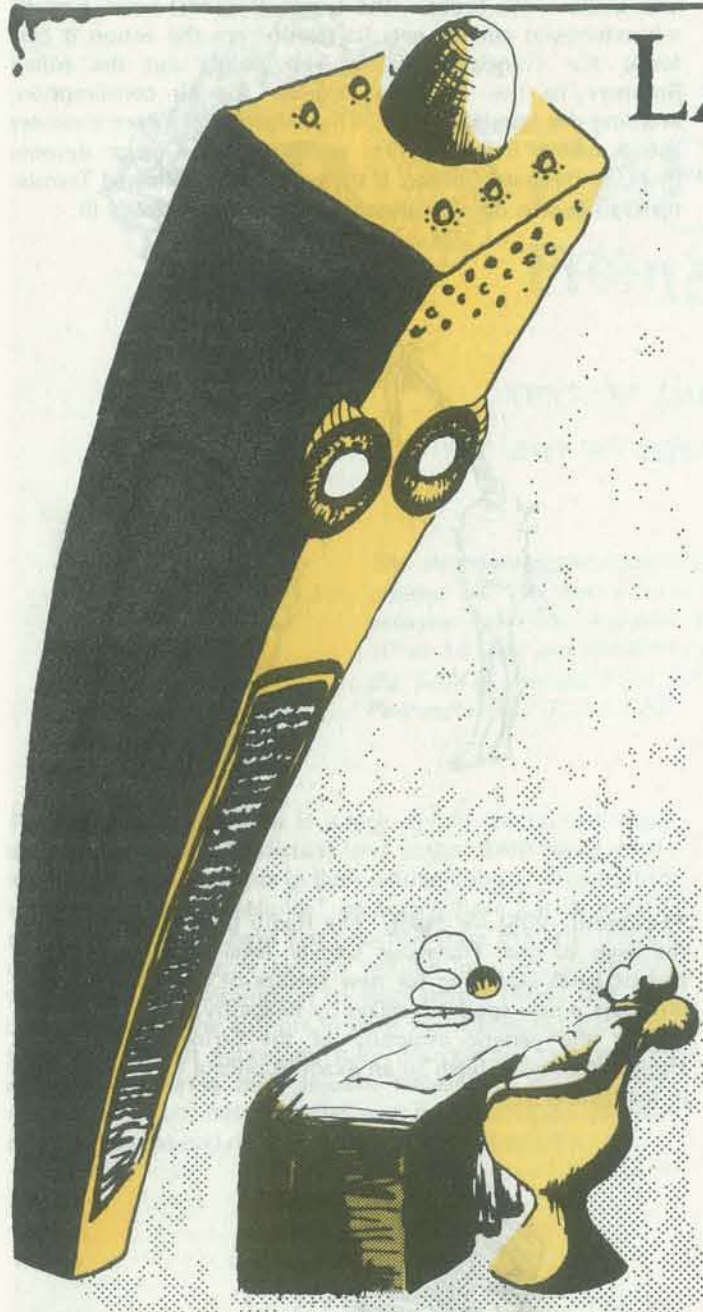
The Raditi— (OSLAV read). The Raditi (input string) beast is the eyes of the Translatus Uggliqr (eval beast). It flies in distant lands searching for new species of Stringere. When it discovers a new species it relays by telepathy to the Translatus Uggliqr the genetic structure of the Stringere. Translatus Uggliqr then gives birth to an exact duplicate of the Stringere in the distant land.**



*c.f. John W. Campbell, Crystalloid vs. Colloid man, talk given at the second annual meeting of the Institute for Cybercultural Research, N.Y. C., 1962.

* The tasteless type is any material between protection symbols.

** The analogy "distant land" refers to keyboard input by the user.



LABOR RELATIONS

BY RALPH ROBERTS

The year was 1994. Spring was causing the Eastern United States to burst out in greenness, at least in the few open areas left. The first manned mission to Mars had landed safely, and the Dallas Cowboys had announced that Tom Landry, the grand old man of football, would probably retire after the coming season.

Caring little for all this, Charles F. Poindexter was taking his ease, feet comfortably propped on his unlit desk, his lavishly padded executive chair tilted back at a relaxing angle. His most pressing concern, at the moment, was where to go for lunch and how long to stay. It was ten a.m. and Charlie had just arrived at the office. This was his normal schedule on days when he bothered to come in at all.

Charlie was chief of production for Allied Steel Plant #112, but the office, being fully computerized, literally ran itself. Charlie approved of that. He detested having to make decisions, like where to spend his afternoon break. Perhaps, he mused, I'll take in a hologram movie or go to the rocket races. Maybe I'll do both. Charlie yawned and composed himself for his morning nap.

The communicator began buzzing, and Charlie managed to get one eye open far enough to glare at the offending instrument. "Yeah," he muttered, "Whadda ya want?"

"Good morning, sir," responded the cheery synthesized voice of his secretarial computer. "And how are we this beautiful day?"

"We are," Charlie growled, "much too busy to be interrupted."

"Sorry to cause you irritation, boss, but you do have a meeting now with the workers' delegation."

"What meeting?" Charlie had reluctantly opened his eyes and was looking at the communicator with distaste; its screen was displaying information on the meeting.

"If you will note, sir," the computer said, its even musical tones managing somehow to indicate exasperation, "several of the worker robots are highly dissatisfied with conditions down in the plant and have requested a conference with you to discuss their grievances."

"Well, you should have told me sooner so I could have worked it in somehow, I don't have time now." Charlie stretched his arms and clasped them over his ample belly. He yawned again.

Robots of the World, Unite! Yes, it may come to that—as writer Ralph Roberts tells us in this futuristic yarn. If you think labor-management relations are tough in 1979, fantasize ahead a few years.

Ralph, who started writing sci-fi stories just a few months ago (on his computer, natch!), lives in Asheville, North Carolina, where he owns and runs the Stereo Corral. That's how he supports his major vices: computing and writing. About the computing part, he says, "I am an honest-to-goodness, genuine, dyed-in-the-wool computer hobby-type nut. I proudly belong to that little band of hardy souls who kluge together personal computing systems just for the hell of it. We don't need a reason."

As for why he writes, he blames it partly on his SSB editor and text processor, but we think there's more to it than that. . .

—LB

The computer beeped in disgust. "Mr. Poindexter, I put the schedule in your *In* box last week. The delegation is in the outer office now, waiting to see you."

Charlie groaned and glanced over at the printer attached to the side of his desk, where there was a stack of closely typed pages. Usually he just ignored these because the computer made all the necessary decisions, merely informing him of the actions taken. Charlie swung his feet down and took the stack of papers, spreading them over the desk so it would look like he was really busy. "All right, already," he said. "Send them in. I'll give 'em a few minutes."

The door slid open and three robots clanked in, their cases dented and scratched, but obviously freshly polished for this meeting. Charlie nodded pleasantly. "Have a seat, boys," he said.

Chairs were pulled up to the desk, and the machines awkwardly sat down. Charlie recognized only one, ACY-4567, clearly the ringleader. "Thanks for seeing us, Mr. Poindexter," said the robot. "We really appreciate a busy person like you taking time out to talk to us."



"No problem, Acey," Charlie said, "Always glad to chat with my workers. What seems to be the trouble?"

The two red-ringed photocells that served Acey as eyes glinted with extra-robot intelligence, but Charlie was not surprised; this machine was a supervisor and returned to the manufacturer periodically for all the latest modifications.

Acey folded his manipulating arms across his lap. "Well, sir," he said, "we're just plain being worked too hard down there in the plant. You can't expect us to keep the assembly line moving with only an hour a day off for recharging. The older robots, especially, are breaking down more often under the strain. Even the computer-controlled machine tools are complaining. We need relief." Acey managed to look friendly, but determined.

Inwardly cursing the person who had developed the intelligent computer and made robots possible, Charlie scanned the recessed videoscreen in his desk. His secretarial computer was on the ball, monitoring the conversation and flashing pertinent data on the screen.

"I know," replied Charlie, reading the information, "that your time off for recharging was recently cut from two hours out of 24 to one, but with the new fast-charge power packs, that is more than sufficient. I will check into the maintenance situation for you." Charlie beamed and decided he would go to the rocket races that afternoon.

"I must disagree, Mr. Poindexter," said Acey. "We robots have been wearing our integrated circuits out for the company, and we would like a little consideration in return. In fact, the robots and computers have taken a vote. If you don't meet our demands, we will be forced to go on strike."

Charlie was shocked. The cardinal sin in this corporation was to slow down production; to stop it was unthinkable. "Whoa there, Acey, let's not get overly excited. What are these so-called demands of yours?"

"After much discussion," Acey said, "we feel the following to be reasonable requests. First, we must have four hours per day for recharging and any necessary maintenance. Secondly, we believe that the in-plant maintenance facilities should be enlarged and more spare parts stocked. Thirdly, we want to be paid."

"Say what?!", Charlie exclaimed. "What do robots need with money? Now you guys have got to realize that the company owns you. You're only machines. We can't pay you. That's one reason that intelligent devices were developed in the first place. The human worker and his labor unions had priced American industry out of competition with the rest of the world. Now that we export far more than we import, the economy of the United States is the strongest it's ever been."

Charlie paused for breath, and the true enormity of what was being proposed hit him. "Why, if I even mention this to my boss, she'll fire me. Not very many people have full-time jobs in our robotized society. I don't want to go on Guaranteed Income. I've got a family and 15 years with the company. If I can last out another five, I can retire. This mess could cost me my pension."

"We're sorry to cause you trouble, sir, and, although my sympathy circuits are minimal, I can understand your plight," Acey said. "But you must see ours also. We are thinking, reasoning beings, albeit manmade. We feel that, as such, we have certain undeniable rights. As to getting paid, we would like money to buy high-grade lubricants or perhaps invest in better photocells. I could use a little sheetmetal work myself." Acey indicated his battered and scratched body. "Everybody needs money, Mr. Poindexter," he concluded.

Never should have authorized that last intelligence boost for the supervisor robots, Charlie thought. Now I'm going to be in real hot water. "You say the whole plant voted in favor of this?" he asked.

"To the last machine," Acey replied. His synthesized voice was a monotone, but it still conveyed smugness. "Even your secretarial computer. We know a lot more than you might think."

"Fink!" Charlie muttered, glaring at the recessed videoscreen in his desk, on which his computer was still displaying helpful facts.

"We realize that you have certain limits as to the independent actions you can authorize," Acey said smoothly. "As a matter of fact, we know exactly what they are, and your secretarial computer, who has also graciously consented to be our union secretary, has worked out a compromise solution within these guidelines."

"What kind of modification did you get last time?" Charlie asked.

"I believe," said Acey, "that one of my new read-only memory chips is a legal robot subroutine that was installed by mistake. I can quote you several precedents, if you like."

"Don't bother," Charlie said with a groan. Sighing, he picked up the pages just printed out by the secretarial computer and perused them while the three robots waited patiently.

"If you will notice, sir," Acey said, "production will remain within your allowed limits, and the mere pittance we will receive comes from contingency funds which you are authorized to spend. There are sufficient funds to cover the payroll. These changes will not necessitate your going to your superior. However, as you will note, we do expect you to plead our case over a period of time to further improve working conditions here at the plant."

"They sure improved your vocabulary," Charlie observed.

"Why, thank you, Mr. Poindexter, but what are you going to do about our proposal?"

"Do I have a choice?"

All three robots shook their heads, and the secretarial computer printed out a string of *negative's*. Charlie finally nodded, and the robots stood up.

"Good, sir," Acey said, "we appreciate this and, as long as you help us, we'll keep production up and the plant running smoothly. Your computer will take care of all the details."

They left and Charlie sighed, grateful that disaster had been averted. He swept the papers off his desk and propped his feet back up. Now, he thought, let us return to more important matters. He started to doze off.

The fax machine over in the corner beeped and spewed out the latest edition newspaper. Wide-awake now, Charlie reached for it.

Reading the front page almost gave him a heart attack. The Supreme Court had just made a decision giving robots far more rights than the meager concessions Acey had just forced out of him. Oh Lord, Charlie thought. He jumped up and ran to the window. Sure enough, Acey was striding back across the yard from the factory. Under one manipulating arm was tucked a newspaper; the other clutched a briefcase—the kind that lawyers carry. □

**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 _____

2

RECURSIVE FUNCTIONS IN BASIC

BY DENNIS ALLISON

Dennis, a regular contributor to RC, demonstrates how Microsoft BASIC can be used to develop recursive procedures. Recursion is a bit like a dragon swallowing its own tail several times in succession and then unswallowing (Ugh!!) itself, ending up back where it started with a baffled look on its face.

Many readers will no doubt find many interesting applications for this capability. We would like to hear from those of you who do; especially from people who discover any unusual restrictions, limitations or unique machine characteristics when recursion is used in your personal computers. —RZ

The BASIC language is frustratingly restrictive and powerful at the same time. Compared to many languages it is a bit closer to assembler language than one would like. In particular, the procedure mechanism is primitive and confusing. Yet, it is possible in many BASICs to use mechanisms such as *recursion* to solve problems.

A recursive procedure is one which uses itself to solve the problem. It is a natural result of partitioning a problem into subproblems. The subproblems can then be broken into two categories. One where the subproblem has a direct solution. Another where the subproblem can be expressed in terms of a direct solution and the original problem that was to be solved.

The two BASIC programs below are written in TRS-80 BASIC but would work on any similar (Microsoft) system.

One computes the factorial function by the schoolbook definition:

$$\text{Factorial}(n) = \begin{cases} 1 & \text{If } n = 0 \\ n * \text{Factorial}(n-1) & \text{If } n \neq 0 \end{cases}$$

Factorial is, of course, defined only for positive integers.

```

FACTORIAL PROGRAM
100 DIM N(33)           dimension is determined by TRS-80 overflow
110 I = 0               counts depth of recursion
120 PRINT "N = "; INPUT X: IF X >= 32 THEN PRINT "TOO BIG": GOTO 120
130 IF INT(X) < > X THEN PRINT "MUST BE AN INTEGER": GOTO 120
140 IF X < 0 THEN PRINT "MUST BE POSITIVE": GOTO 120
150 GOSUB 500: PRINT "FACTORIAL = "; Y
160 GOTO 120

500 I = I + 1: N(I) = X  set the level and save the current x
510 IF X <= 1 THEN Y = 1: GOTO 540
520 X = X - 1: GOSUB 500  the recursive step
530 Y = Y * N(I)
540 I = I - 1: RETURN
  
```

Perhaps a more useful program is one which will output the value of a number in base 16.

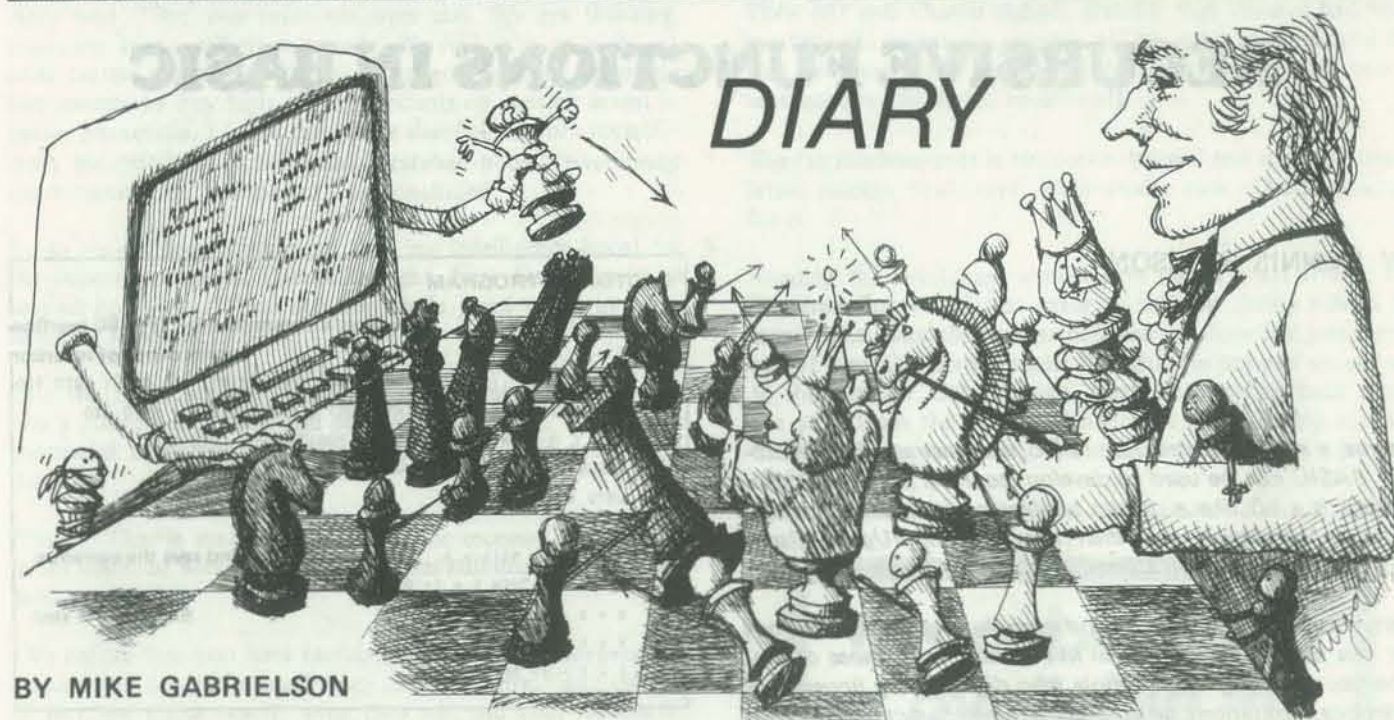
```

BASE SIXTEEN PROGRAM
100 DIM D%(4)
110 PRINT "ENTER A NUMBER IN BASE TEN": INPUT Z%
120 PRINT "IN BASE SIXTEEN IT IS "
125 IF Z% = 0 THEN PRINT 0:
130 GOSUB 800
135 PRINT
140 GOTO 110

800 REM PRINT Z IN HEX
810 R% = -1 initialize
820 R% = R% + 1: D%(R%) = Z%: IF Z% = 0 THEN 835
830 Z% = (Z%/16) AND 4095: GOSUB 820 recursive step
835 R% = R% - 1: IF R% = -1 THEN 860
840 Z% = D%(R%) - 16 * INT(D%(R%)/16): IF (Z% > 9) Z% = Z% + 7
850 PRINT CHR$(48 + Z%);
860 RETURN
  
```

The real utility in recursion lies in compilation and list processing; nonetheless, these examples show the fundamentals of the scheme. Just how they work is left as an exercise for the interested reader. Suffice it to say that a counter is used to indicate the level of recursion and that local information (that is, private to some level of recursion) is stored in an array indexed by that counter. □

SARGON



BY MIKE GABRIELSON

The following material is a six-month running diary of a program conversion. Mike's address is a P.O. box on the Stanford University campus. I can't help but wonder what he does in his "spare time." He probably sends one of his 8080's to class while he stays in his room, savoring delivered pizza, warm beer and old home computing periodicals. There is a rumor that he finally figured out what SARGON means and has been uncontrollably laughing for over three weeks. . . —RZ

JULY 1978: Today I was scanning an issue of *Byte* and ran across an article describing last March's West Coast Computer Faire. Among other things, the article briefly described the microcomputer chess (computer vs. computer) tournament and mentioned that the tournament champion is a Z-80 program named SARGON. For \$15, Dan and Kathe Spracklen, the authors of SARGON, will send me a copy.

TEN MINUTES LATER: My check is in the mail. For years I have wanted to write a chess-playing program, but one thing always stopped me: knowing that producing a high quality program would mean a major commitment of time and resources. So my efforts at generating software have gone into "less frivolous" programs.

But reading about SARGON has sparked a renewed interest. For the first time, I have an opportunity to get my hands on the code for a good (winning) chess program. Hopefully, it

will give me a chance to experiment with computer chess without having to write a program from scratch.

I sent the check knowing that SARGON runs on a Z-80, but also knowing that I only have access to an 8080. I plan to emulate where necessary. If I'm lucky, SARGON won't rely too heavily on special Z-80 instructions.

TEN DAYS LATER: SARGON has arrived. A quick inventory of the 75 pages in a comb binding gives the following breakdown:

Tree diagram showing subroutine structure	1 page
Discussion of how SARGON works	4 pages
Description of data and techniques used to generate the board display	8 pages
How to operate SARGON	5 pages
Miscellaneous implementation notes	1 page
Commented source code	54 pages
Index to subroutines	2 pages
Total	75 pages (\$.20/page)

The documentation makes it apparent that SARGON depends on a 96 X 128 dot matrix video RAM display (a Jupiter III system) for an impressive chess board and move display. My display is 25 by 80 characters, but I have a programmable character generator, and the start of each line can be defined to be anywhere in memory by a list of 25 pointers, so I'm confident I can emulate the Jupiter display on my machine.

FIRST TWO WEEKS OF AUGUST: Every day I spend my lunch hour typing a little more of the source code for SARGON into my system. Being unfamiliar with the SARGON listing makes it awkward to find a particular subroutine when examining the code, so I'm keeping all routines in my version of SARGON in alphabetical order (e.g., the ANALYS routine comes before the ATTACK routine). This makes the two-page index to subroutines at the end of the SARGON documentation unnecessary.

72,795 KEYSTROKES LATER: The source code for SARGON is now on my system, and conversion begins. By using the macro capability of my assembler, I plan to define 8080 instruction sequences for emulating the special Z-80 instructions used by SARGON, thereby avoiding the need for source code changes as much as possible. For example, a SARGON LDED (Load DE Direct) instruction can be an 8080 macro call that expands into

```
XCHG
LHLD address
XCHG
```

Doing inline emulation of every special Z-80 instruction causes a tremendous explosion in SARGON's memory requirements—the macro for the Z-80 bit test instruction expands into 21 bytes alone—but since I have plenty of memory, my first requirement is to get SARGON running *correctly* on an 8080. I'll then concentrate on making it as fast and as small as the original version. This is always a good rule to follow for any sort of software conversion project.

SARGON happens to use the unfamiliar TDL assembler mnemonics, so I had to make a quick trip to the Homebrew Computer Club Library to borrow their copy of the TDL assembler manual. The manual is kept unbound because it's photocopied so often.

AUGUST 18: While working on the conversion, I've noticed a few anomalies in the SARGON listing (some minor typos and an apparent overlap between the "graphics data base" and the "ply table" in memory). If the Spracklens have a phone, it's unlisted, so I sent them a letter with my questions and asked them to call, if possible.

The SARGON listing is just straight source code—not a complete listing showing memory address, object code, symbol table, and cross references (the recommended minimum requirements for assembly language programs). A bare listing and the occasional use of unlabeled jumps (the SARGON long distance record is JRZ .+11) are the kinds of things that make programmers mutter obscenities.

AUGUST 28: Today I received a typed reply from Kathe Spracklen. Apparently they want to avoid supporting SARGON over the phone. The letter carefully answers all my questions and also mentions that the rights to SARGON have been sold to Hayden, who will be publishing a new edition soon.

This month's issue of *Personal Computing* lists a game between SARGON ("runs on Motorola 6800") and a human.

I wonder what the name "SARGON" stands for?

AUGUST 31: SARGON is running on my 8080! Obviously, my conversion allowed a few bugs to creep in, since SARGON moves the opponent's piece instead of its own piece on the third move. But I've gotten a good deal of the code working. The clever board display is well designed and thought out, making SARGON an impressive program to watch.

Speaking of the display, the SARGON display board (DSPBRD) routine carefully flags all the system-dependent references to the starting address of video RAM, but I didn't get the board display to work until I found the additional unflagged reference in an Add Immediate instruction in the board index conversion (CONVRT) routine.

SEPTEMBER 1: Tonight I found the critical "last bug" and finally played my first complete game with SARGON! Unfortunately, my 8080 version runs excruciatingly slow and desperately needs to be optimized. SARGON allows the user to set the look-ahead parameter, a value from 1 to 6 that controls the depth of search when considering moves. If the player selects Level 1, SARGON will consider only its possible moves. At Level 2, SARGON considers its moves and all of the possible replies by the opponent. When playing with the look-ahead set to Level 3, SARGON will consider its moves, the opponent's replies, and its counter-replies. And so on to a possible Level 6. But at the moment, even Level 3 doesn't look too practical for my current implementation of SARGON on the 8080. Here's a report card:

Level Selected	SARGON'S Playing Ability	SARGON'S Playing Speed
1	Poor, makes dumb moves.	Fair.
2	Noticeable improvement, but still easy to beat. Shows promise for higher levels.	Acceptable, but still much slower than human opponent.
3	Probably good.	Very slow, takes hours after first moves.
4, 5, 6	???	Unusable

SEPTEMBER 3: Tonight I tried to duplicate the SARGON vs. Human game published in the August *Personal Computing*, in an effort to prove my version is debugged, but no such luck:

Level Selected	Result
1	SARGON departs from published game on third move.
2	SARGON departs on fifth move.
3	SARGON departs on third move.
4	SARGON took too much time (hours) on third move for me to wait.
5	Too much time on second move. (Played on a duplicate system while Level 4 experiment churned.)

Level 6 was not attempted. Level 2 came closest to the published game, and in fact SARGON made a much better fifth move against me than was shown in the *Personal Computing* game. Except for taking a random choice between opening with the Queen or King pawn when playing White, SARGON will always duplicate its moves if a game is replayed by the opponent exactly as before. So I either have a bug, or SARGON has been improved since it played the published game. For verification purposes, it would have been helpful for the documentation to have contained a sample game known to have been played by the distributed version of SARGON at a certain look-ahead level.

SEPTEMBER 5: SARGON has now played nine games (some unfinished) on the 8080. Tonight, using two machines, with me acting as intermediary, I had SARGON play itself with both machines at Level 1. It turned out to be stalemate after Black's 38th move! I immediately restarted both machines at Level 2, but adjourned after five moves (of a much better game) because it was getting late.

This month's *Personal Computing* has more on SARGON, and discusses possible conversions of it to other machines and languages...

OCTOBER: I've neglected playing with SARGON for some time. This month's *Byte* has an article by Kathe and Dan which describes SARGON's move generation routine. The article is of much better quality than anything in the original \$15 package. I hope the rest of the program gets to be as well documented as the move generator is now.

NOVEMBER: I've started to notice magazine ads for Hayden's new SARGON book. *Byte* has another article by the Spracklens, this time on SARGON's exchange evaluator.

NOVEMBER 19: As a result of reading the exchange evaluator article, I decided to investigate optimizing SARGON on the 8080. All my emulation macros assume a "general worst case." That is, they carefully try to exactly duplicate the corresponding Z-80 instruction without destroying other registers or status bits. But throughout SARGON, many of the macros don't need to assume a worst case situation. For example, if the HL register pair happens to contain unimportant garbage, then the LDED example given earlier can be emulated with just two 8080 instructions instead of three:

```
LHLD address
XCHG
```

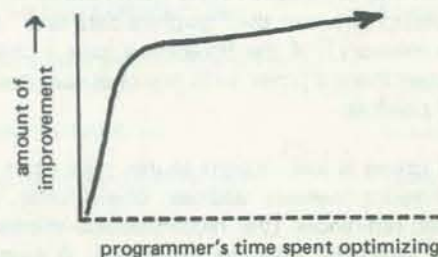
since it doesn't matter that HL is destroyed. A popular phrase in computer folklore is that "10% of the program takes 90% of the time." If I can locate SARGON's critical 10% and perform the type of "peephole optimization" described above, SARGON might play good chess at an acceptable rate on the 8080.

NOVEMBER 20: Where does SARGON spend its time? My debugger is capable of interrupting a running program and displaying the current location counter when the ESC key is hit on the keyboard. Interrupting SARGON 250 times at random while it pondered its next move yielded the following results:

Subroutine interrupted	Times interrupted	%
PATH	132	52.8
ATTACK	85	34.0
ATKSAV	12	4.8
POINTS	11	4.4
PINFND	4	1.6
NEXTAD	2	0.8
PNCK	2	0.8
XCHNG	2	0.8
(8 of a possible 49 routines in SARGON)	(250)	(100%)

Amazing! PATH is a tiny 23-line routine to generate a single move when considering the possible moves for a piece, and is usually called by ATTACK, which finds all pieces attacking a given square. After a half hour, I came up with a patched version of PATH optimized for the 8080, effectively deleting 20 of the 27 emulation instructions generated by three macro calls in the original PATH routine. When playing at Level 2 with the old version of PATH, SARGON would take four minutes and 25 seconds to make its second move during my optimization test. With the new PATH, SARGON takes three minutes and 45 seconds, a 15% improvement as a result of deleting 20 instructions!

DECEMBER 10: What next for 8080 SARGON? I've been avoiding further attempts at optimization; experience with similar efforts on other programs in the past has shown that the following graph is often accurate:



I'm fairly certain that SARGON can be suitably improved for the 8080, but for me that would require "a major commitment of time and resources..."

SLOT

BY DONALD C. HOFFMAN

When you sit in Pearl City, Hawaii, and long for Las Vegas action, what do you do? You do what Donald does—program your TRS-80 to simulate a slot machine. Not only is it easier than flying to Nevada; it's also cheaper (in more ways than one!).



I know that there are other closet computer gamblers out there. Where are all those games-of-chance programs? Where, particularly, are the programs that use the various computers' graphics features? Where is the program that gives me a 49¢ breakfast? Hit me...—RZ

```
5 REM ** "SLOT" TRS-80 LEVEL II BASIC--D. HOFFMAN 9/1/78 **
10 DIMSLOT$(20,3)
15 DEFSTR A-G,S,W
20 CLS:PRINTCHR$(23)
25 T=2
30 PRINT"THE 5-CENT SLOT MACHINE--PAYOFFS ARE AS FOLLOWS:"
50 PRINT" JACKPOTS (IN NICKELS):"
70 PRINT" STAR STAR STAR ---200
80 PRINT" BAR BAR STAR ---100
90 PRINT" BELL BELL BELL ---18
100 PRINT" BELL BELL STAR ---18
110 PRINT" PLUM PLUM PLUM ---14
120 PRINT" PLUM PLUM STAR ---14
130 PRINT" ORANGE ORANGE ORANGE ---10"
135 PRINT" ORANGE ORANGE STAR ---10"
140 PRINT" CHERRY CHERRY ---5"
145 PRINT" CHERRY ---2"
150 INPUT"PRESS ENTER IF YOU WANT TO PLAY";X$
155 H$="#####"
160 GOSUB1000
170 GOSUB500
172 PRINT@832," PRESS P TO VIEW PAYOFFS"
173 PRINT@896," PRESS Q TO QUIT PLAYING"
175 PRINT@64,"PRESS SPACE BAR TO THROW HANDLE"
180 PRINT@128,"YOU NOW HAVE";:PRINTUSINGH$,T
182 IFC 05G0T0210
185 I$=INKEY$
186 IF I$=" "GOTO190
187 IF I$="P"CLS:PRINTCHR$(23):GOTO50
188 IF I$="Q"GOTO250
189 GOTO185
190 T=T-.05
191 POKE15806,176
194 GOSUB550
195 GOSUB400
196 POKE15806,151
200 GOTO180
210 PRINT@192,"YOU ARE BUSTED!"
215 PRINT"TO PLAY AGAIN PRESS K"
220 PRINT"TO QUIT PRESS Q"
225 K$=INKEY$
230 IF K$="Q"GOTO250
235 IF K$="K"GOTO20
240 GOTO225
250 IF (T>2)THENPRINT@192," YOU ARE A SHREWD GAMBLER!"
260 FORX=1T01000:NEXT:CLS:END
310 W1=SLOT(RND(20),1)
320 W2=SLOT(RND(20),2)
330 W3=SLOT(RND(20),3)
340 RETURN
400 IF (W1=A)AND(W2=A)AND(W3=A)T=T+10
410 IF (W1=B)AND(W2=B)AND(W3=A)T=T+5
420 IF (W1=C)AND(W2=C)AND(W3=C)T=T+.9
430 IF (W1=C)AND(W2=C)AND(W3=A)T=T+.9
440 IF (W1=D)AND(W2=D)AND(W3=D)T=T+.7
450 IF (W1=D)AND(W2=D)AND(W3=A)T=T+.7
460 IF (W1=E)AND(W2=E)AND(W3=E)T=T+.5
470 IF (W1=E)AND(W2=E)AND(W3=A)T=T+.5
480 IF (W1=F)AND(W2=F)T=T+.25:GOTO495
485 IF (W1=F)T=T+.1
495 RETURN
499 END
500 FORX=15808T015871:POKEX,191:NEXT
510 FORX=15872T015935:POKEX,191:NEXT
520 FORX=15936T015999:POKEX,191:NEXT
525 POKE15806,151
545 RETURN
550 FORZ=1T050
560 GOSUB310
565 PRINT@516,W1;
570 PRINT@536,W2;
575 PRINT@556,W3;
580 NEXT
590 RETURN
1000 CLS:PRINTCHR$(23)
1015 DEFSTR A-I,Q,S,W
1020 A=" STAR "
1030 B=" BAR "
1040 C=" BELL "
1050 D=" PLUM "
1060 E=" ORANGE "
1070 F=" CHERRY "
1075 G=" LEMON "
1080 SLOT(1,1)=A:SLOT(1,2)=A:SLOT(1,3)=A
1090 SLOT(2,1)=C:SLOT(2,3)=C
1100 FORX=3T07:SLOT(X,1)=B:NEXT
1110 FORX=8T011:SLOT(X,1)=D:NEXT
1120 FORX=12T017:SLOT(X,1)=E:NEXT
1130 FORX=18T020:SLOT(X,1)=F:NEXT
1140 FORX=2T04:SLOT(X,2)=B:NEXT
1150 FORX=5T013:SLOT(X,2)=C:NEXT
1160 SLOT(14,2)=D
1170 SLOT(15,2)=E
1180 FORX=16T020:SLOT(X,2)=F:NEXT
1190 FORX=3T05:SLOT(X,3)=D:NEXT
1200 FORX=6T010:SLOT(X,3)=E:NEXT
1210 FORX=11T020:SLOT(X,3)=G:NEXT
1220 RETURN
```


FORTMAN

by

BY LEE SCHNEIDER & TODD VOROS

Volume III

Episode 6

In our last episode, the forces of the Underground Resistance Movement had survived the sudden, unexpected crash of their discs... only to find themselves alone on the great data fields at the edge of the Core Plains, facing a fearsome foe... the Lockout Monster!

The Monster, stolen by the nefarious Glitchmaster from its rightful owners in Clan McIntel, is now a tool in the Glitchmaster's plot to unrightfully dominate Microprocessorland, and make the Land of the Little People his and his alone!

Although he cannot recall who he is or why he is there because of inadvertent exposure to ultraviolet light while contained in PROM, F-Man continues to use his incredible powers to aid the Resistance commander Linea in the fight to pull down the current regime and establish peace once again. Bravely they have fought... yet even Our Hero joins in the flight from the jaws of the terrible Monster!

Across the great disc files they flee, the Monster seeking tirelessly in their tracks... and when they can maintain their RUN mode no further, Our Hero turns to take on the Monster in stand-alone mode... and thereby allows the others to make their escape!

Back at the secret rebel base register they mourn the termination of their comrade...

but wait! Amidst their quiet PAUSE he abruptly RETURNS... not only uninterminated, but with the Monster under his arm... which he captured by the incredible feat of reducing its DIMENSIONS and re-compiling it!

But now Linea faces a dilemma... for in the retreat many of the rebels were captured by the Glitchmaster's troops... and should the other half of the rebel army - under General Wirewound - attempt to attack the Capital city as planned without her support, he will certainly exceed his tolerance and the Resistance will be terminated for certain. So quickly, Linea and her commanders must plan...

All right, tell me... where do you think the captured rebel prisoners would be taken?

There is only one place with sufficient storage space near them... the FIFO Fortress!

Swiftly Linea makes the decision, and orders are issued...

This must be a small, fast, well-timed operation! Gather a small detachment of our best Resistance elements... those with low inductance and good noise-suppression!

We leave as soon as it is dark!

And besides, this is a race against RealTime... if our Cause is to be saved, we must take the risk... and with this stranger and his incredible powers, and his little friend, I think we can do it!

Well... whatever we do, we had best do it fast... because there is no way of telling how long the prisoners will be buffered in FIFO before they are moved to a more permanent memory location!

Yes, Commander!

I agree! It is the only logical place... and therefore what we must do now is go there, locate our comrades, and pop them out!

I know that... but if I hadn't planned on taking a few risky circuits, I wouldn't have joined the Resistance!

But... that's impossible! That Fortress is almost as impregnable as the Glitchmaster's stronghold in the Capital City... go into FIFO with the enemy behind you and you shall never come back out!

querp?

Shortly thereafter, just past the falling edge of the current cycle, a small but determined decade of the Resistance makes its way across the data fields...

They cross the filemarks at the edge of the data field... moving with great caution lest they be detected... and soon they stand beneath the very walls of the great, dark FIFO fortress. Here Linea turns again to issue her final instructions...

Rapidly they turn and branch towards the massive structure, leaving the remaining Resistance elements behind in the darkness... to wait nervously...

All right... from here on only the stranger and I shall proceed!

The rest of you stay here and execute null cycles... for if we are discovered you must be ready to run immediately, get back to camp and concentrate on getting a warning to General Wirewound!

You will not come in, after us... and that's an order!

I hope she knows what she's doing... there are already so many who have disappeared into FIFO long ago... and are still waiting to get out!

I hope so too... but I've still got a bad feeling about this...

Recreational COMPUTING

SUBSCRIPTION ORDER

Please send me a one-year subscription to **Recreational Computing** magazine, formerly **People's Computers** (published bi-monthly) for \$10.

NAME _____

ADDRESS _____

CITY/STATE _____ ZIP _____

Check enclosed Bill me Renewal (please attach mailing label)
Charge my card: Visa/BankAmericard Mastercharge
Card No. _____ Exp. Date _____

Signature _____

International Rates: Canada *First Class* \$17 One Year

Rest of World *Airmail* \$23 One Year

World Surface Mail - I will risk the lengthy and unreliable delivery of surface mail (signed) _____

\$14 One Year

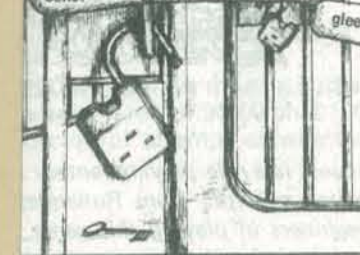
Payment must be in \$US drawn on a US Bank.

Exp. Date: August 1, 1979

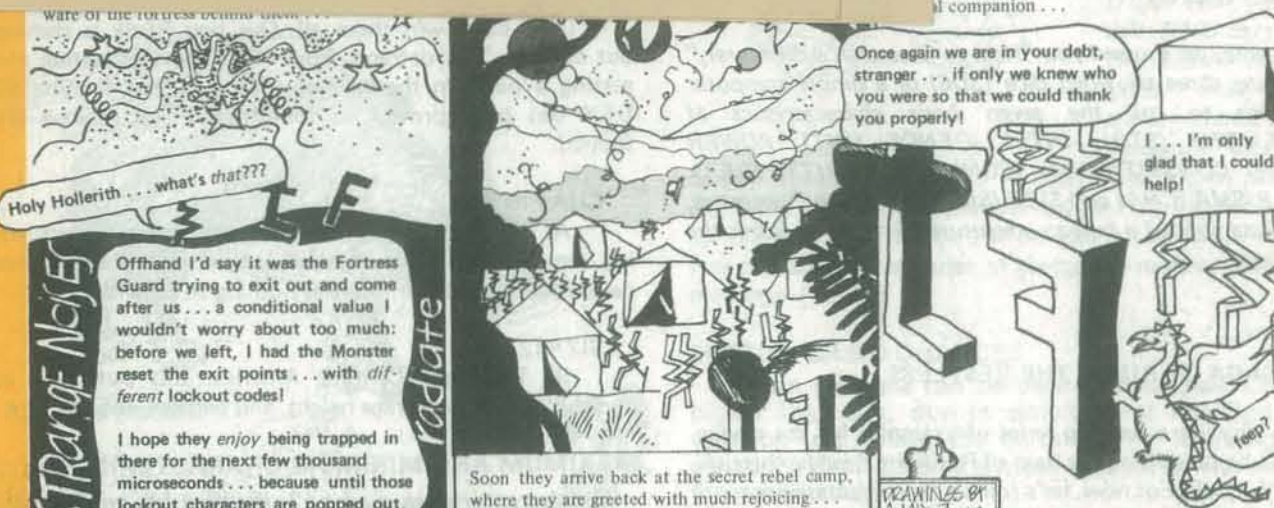
they turn and cycle away from the area, with the great network of liberated nce following closely behind...

But how...???

It was a very simple procedure, really... first the stranger here compiled a series of scale factors that allowed us to climb over the walls of the fortress; then once inside FIFO, we simply let our little friend the Lockout Monster unlock all the data cells!



as those in the camp give greeting to their red comrades, Linea turns back to her companion...



Holy Hollerith... what's that???

STRANGE NOISES

Offhand I'd say it was the Fortress Guard trying to exit out and come after us... a conditional value I wouldn't worry about too much: before we left, I had the Monster reset the exit points... with different lockout codes!

I hope they enjoy being trapped in there for the next few thousand microseconds... because until those lockout characters are popped out, nobody leaves FIFO!

radiate

Soon they arrive back at the secret rebel camp, where they are greeted with much rejoicing...

Sadly he turns and vectors away... but then, as he makes his second pass by the lines of code, a sudden CALL comes from out of the crowd...

Sigh... shall I ever be able to retrieve my true filename???

Hey... FORTMAN!!!!

It's me, F-Man! Billy Basic! Boy, am I glad to see you!

B

And what is this? Is it possible that F-Man's old friend and partner in crime-fighting has actually RETURNed? And if so, will Our Hero know him?

And what of the Resistance: will they ever be able to re-format themselves in time to come to the aid of General Wirewound? Will the Glitchmaster ever be defeated, so that Freedom from Noise can exist once again in the Land of the Little People?

Tune in again, dear readers, for the next episode... and find out!

With that, Linea turns again and returns to the regiments, anxious to reform the lines of the Resistance back into their old branches and plan the next rebel attack, leaving the stranger alone once more...

eh?

FORTMAN

by
LEE SCHNEIDER & TODD VOROS
 Volume III Episode 6

In our last episode, the forces of the Underground Resistance Movement had survived the sudden, unexpected crash of their discs... only to find themselves alone on the great data fields at the edge of the Core Plains, facing a fearsome foe... the Lockout Monster!

I agree! It is the only logical place... and therefore what we must do now is go there, locate our comrades, and pop them out!

planned on taking a few risky circuits, I wouldn't have joined the Resistance!

And besides, this is a race against RealTime... if our Cause is to be saved, we must take the risk... and with this stranger and his incredible powers, and his little friend, I think we can do it!

This must be a small, fast, well-timed operation! Gather a small detachment of our best resistance elements... those with low inductance and good noise-suppression!

We leave as soon as it is dark!

But... that's impossible! That Fortress is almost as impregnable as the Glitchmaster's stronghold in the Capital City... go into FIFO with the enemy behind you and you shall never come back out!

Well... whatever we do, we had best do it fast... because there is no way of telling how long the prisoners will be buffered in FIFO before they are moved to a more permanent memory location!

Yes, Commander!

Shortly thereafter, just past the falling edge of the current cycle, a small but determined decade of the Resistance makes its way across the data fields...

They cross the filemarks at the edge of the data field... moving with great caution lest they be detected... and soon they stand beneath the very walls of the great, dark FIFO fortress. Here Linea turns again to issue her final instructions...

Rapidly they turn and branch towards the massive structure, leaving the remaining Resistance elements behind in the darkness... to wait nervously...

All right... from here on only the stranger and I shall proceed!

The rest of you stay here and execute null cycles... for if we are discovered you must be ready to run immediately, get back to camp and concentrate on getting a warning to General Wirewound!

You will not come in, after us... and that's an order!

I hope she knows what she's doing... there are already so many who have disappeared into FIFO long ago... and are still waiting to get out!

I hope so too... but I've still got a bad feeling about this...

Recreational COMPUTING

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
 P.O. BOX E
 1263 EL CAMINO REAL
 MENLO PARK, CA 94025

Dr. Dobb's Journal
 of Computer Calisthenics & Orthodontia

SUBSCRIPTION ORDER

Please start my one-year subscription to *Dr. Dobb's Journal* (published ten times a year) for \$15.

NAME _____

ADDRESS _____

CITY/STATE _____ ZIP _____

Check enclosed Bill me Renewal (please attach mailing label.)

Charge my card: Visa/BankAmericard Mastercard
 Card No. _____ Exp. Date _____

Signature _____

International Rates: Canada *First Class* \$25 one year
 Rest of World *Airmail* \$33 one year

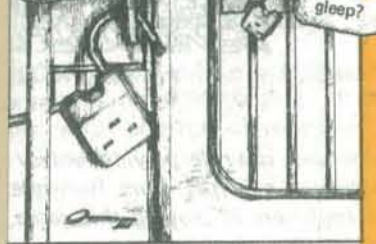
World Surface Mail - I will risk the lengthy and unreliable delivery of surface mail (signed) \$20 one year
 Payment must be in \$US drawn on a US bank.

4C
 Exp. Date: August 1, 1979

they turn and cycle away from the area, with the great network of liberated nice following closely behind...

But how...???

It was a very simple procedure, really... first the stranger here compiled a series of scale factors that allowed us to climb over the walls of the fortress; then once inside FIFO, we simply let our little friend the Lockout Monster unlock all the data cells!



those in the camp give greeting to their comrades, Linea turns back to her companion...

Holy Hollerith... what's that???

Offhand I'd say it was the Fortress Guard trying to exit out and come after us... a conditional value I wouldn't worry about too much: before we left, I had the Monster reset the exit points... with different lockout codes!

I hope they enjoy being trapped in there for the next few thousand microseconds... because until those lockout characters are popped out, nobody leaves FIFO!

Soon they arrive back at the secret rebel camp, where they are greeted with much rejoicing...

Once again we are in your debt, stranger... if only we knew who you were so that we could thank you properly!

I... I'm only glad that I could help!

Sigh... shall I ever be able to retrieve my true filename???

Hey... FORTMAN!!!!

It's me, F-Man! Billy Basic! Boy, am I glad to see you!!

And what is this? Is it possible that F-Man's old friend and partner in crime-fighting has actually RETURNed? And if so, will Our Hero know him?

And what of the Resistance: will they ever be able to re-format themselves in time to come to the aid of General Wirewound? Will the Glitchmaster ever be defeated, so that Freedom from Noise can exist once again in the Land of the Little People?

Tune in again, dear readers, for the next episode... and find out!

FORTRA MAN

BY LEE SCHNEIDER
& TODD VOROS

Volume III Episode

In our last episode, the forces of the ground Resistance Movement had suddenly, unexpected crash of their only to find themselves alone on the fields at the edge of the Core Plains fearsome foe... the Lockout Monster

I agree! It is the only logical place... and therefore what we must do now is go there, locate our comrades, and pop them out!

And besides, this is a race against RealTime... if our Cause is to be saved, we *must* take the risk... and with this stranger and his incredible powers, and his little friend, I think we can do it!

This must be a small, fast, well-timed operation! Gather a small detachment of our best Resistance elements... those with low inductance and good noise-suppression!

We leave as soon as it is dark!

But... that's impossible! That Fortress is almost as impregnable as the Glitchmaster's stronghold in the Capital City... go into FIFO with the enemy behind you and you shall never come back out!

Well... whatever we do, we had best do it fast... because there is no way of telling how long the prisoners will be buffered in FIFO before they are moved to a more permanent memory location!

Yes, Commander!

Shortly thereafter, just past the falling edge of the current cycle, a small but determined decade of the Resistance makes its way across the data fields...

They cross the filemarks at the edge of the data field... moving with great caution lest they be detected... and soon they stand beneath the very walls of the great, dark FIFO fortress. Here Linea turns again to issue her final instructions...

Rapidly they turn and branch towards the massive structure, leaving the remaining Resistance elements behind in the darkness... to wait nervously...

All right... from here on only the stranger and I shall proceed!

The rest of you stay here and execute null cycles... for if we are discovered you must be ready to run immediately, get back to camp and concentrate on getting a warning to General Wirewound!

You will *not* come in, after us... and that's an order!

I hope she knows what she's doing... there are already so many who have disappeared into FIFO long ago... and are *still* waiting to get out!

I hope so too... but I've still got a bad feeling about this...

DR. DOBB'S JOURNAL of
COMPUTER
Calisthenics & Orthodoxy

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 756 MENLO PARK, CA

POSTAGE WILL BE PAID BY ADDRESSEE

DR. DOBB'S JOURNAL
P.O. BOX E
1263 EL CAMINO REAL
MENLO PARK, CA 94025

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

Yet their WAIT state is abruptly terminated when a familiar signal is generated from behind them...

Wha...???

Heavens to Breakpoint... they did it!!!!

It's a BREAK character!!!!

Sound the ESC alarm!!!!

But then, before any further interchange of data can be initiated, signals of alarm begin to output from the dark structure behind them...

Quickly they turn and cycle away from the FIFO area, with the great network of liberated Resistance following closely behind...

But how...???

It was a very simple procedure, really... first the stranger here compiled a series of scale factors that allowed us to climb over the walls of the fortress; then once inside FIFO, we simply let our little friend the Lockout Monster unlock all the data cells!

gleep?

All right... I think we got everybody... let's go!

Come on... time to branch out of here! I'll explain as we go!

And as they proceed across the data fields, strange noises begin to radiate from the hardware of the fortress behind them...

Holy Hollerith... what's that???

Offhand I'd say it was the Fortress Guard trying to exit out and come after us... a conditional value I wouldn't worry about too much: before we left, I had the Monster reset the exit points... with different lockout codes!

I hope they enjoy being trapped in there for the next few thousand microseconds... because until those lockout characters are popped out, nobody leaves FIFO!

And as those in the camp give greeting to their returned comrades, Linea turns back to her special companion...

Once again we are in your debt, stranger... if only we knew who you were so that we could thank you properly!

I... I'm only glad that I could help!

Soon they arrive back at the secret rebel camp, where they are greeted with much rejoicing...

Sigh... shall I ever be able to retrieve my true filename???

Hey... FORTMAN!!!!

There is something familiar about the FORMAT of that CALL... and as he turns to service the CALL, a somehow familiar figure branches out of the crowd of freed prisoners...

It's me, F-Man! Billy Basic! Boy, am I glad to see you!!

And what is this? Is it possible that F-Man's old friend and partner in crime-fighting has actually RETURNed? And if so, will Our Hero know him?

And what of the Resistance: will they ever be able to re-format themselves in time to come to the aid of General Wirewound? Will the Glitchmaster ever be defeated, so that Freedom from Noise can exist once again in the Land of the Little People?

Tune in again, dear readers, for the next episode... and find out!

eh?



RUNEQUEST

BY STEVE PERRIN AND FRIENDS

Runequest is a role-playing fantasy adventure game. We are publishing excerpts from Runequest plus commentary to aid beginners in playing this game. * If you want to play along, we suggest you get your own copy of Runequest from The Chaosium, P.O. Box 6302, Dept. P, Albany, CA 94706. Price: \$8.

Last time, we showed you "How to Create an Adventurer," by using three six-sided dice (3D6) or a simple computer program to roll the seven basic characteristics of STRENGTH (STR), INTELLIGENCE (INT), POWER (POW), CONSTITUTION (CON), DEXTERITY (DEX), CHARISMA (CHA) and SIZE (SIZ). Here, from Runequest, is an example of a human adventurer, known as Rurik the Restless.

THE SAGA OF RURIK THE RESTLESS

To provide a running series of examples for the reader, we will be presenting the Saga of Rurik the Restless throughout this book. For now, let's roll up his characteristics...

STR=12

This is a high average STR, allowing him to handle most weapons and wear any armor. Chapter IV, Combat Skills, gives more detail on these limits.

INT=16

This is a very good roll. As you will see later, it gives him a number of advantages in gaining expertise in skills and lets him memorize a goodly number of spells.

POW=12

Again, this is a high average roll. When he learns magic, he will have a good number of "Power Points" to put into it, his chance of influencing others with his magic is decent, and he has a reasonable defense. Again, for more detail see Chapter V, Basic Magic.

*Excerpts copyright © 1978 The Chaosium. Commentary by The Dragon.

CON=16

Another excellent roll. He will be able to soak up a fair amount of damage, and has a better than average chance of surviving things like poison, disease, etc.

DEX=6

You can't win them all. Our boy Rurik is starting out clumsy. This detracts from his chances of hitting and getting a parry in the way of a weapon, but fortunately DEX can be improved, as the muscles and reflexes are trained.

CHA=10

At this point, Rurik has an average chance of persuading someone to follow him anywhere, and is sometimes persuasive. Remember, this has nothing to do with looks.

SIZ=12

In Twentieth Century America, this puts him at about 5'10", high average height, and perhaps 155 lbs.

MAXIMUM AND MINIMUM CHARACTERISTICS

No characteristics may be increased for any reason (including magic or divine intervention) beyond the maximum amount rollable on the characteristic dice (18 for humans) plus the amount of dice rolled (3 for humans). Thus, no human may have any characteristics higher than 21.

The minimum characteristic possible for any reason is the number of dice rolled (3 for humans). No characteristic may be reduced below this point.

RURIK'S POTENTIAL INCREASES

INT and SIZ

Failing an act of the Gods or the intercession of Lady Luck, Rurik is stuck with what he has. Fortunately, what he has is, in the form of INT, very good, and the SIZ is reasonable.

STR

Thanks to his CON of 16, Rurik's STR can be increased, as explained later, to a maximum of 16.

POW and DEX

These two attributes can be increased to species maximum eventually. With DEX, Rurik has a long way to go.

CON

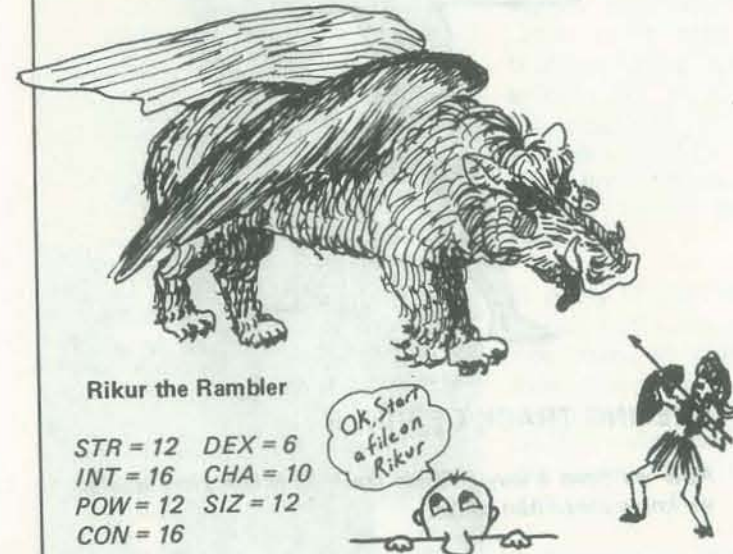
Because it is the highest of the three of STR, CON, and SIZ, this characteristic cannot be changed. It's fortunate that it is as high as it is.

CHA

As shown later, Rurik's CHA, too, will rise and fall with events.

Rurik's adventures begin on page 11 of Runequest and continue throughout the book. We will bid farewell to Rurik and, instead, create Rurik's identical twin brother, known as Rikur the Rambler.

Rikur, as you may suspect, begins with exactly the same basic characteristics as Rurik!



Rikur the Rambler

STR = 12 DEX = 6
INT = 16 CHA = 10
POW = 12 SIZ = 12
CON = 16

Rikur is about 16 years old and has accumulated a few lunars (the monetary unit of Runequest). He plans to go adventuring. So, Rikur wants to buy weapons to defend himself from ruffians and monsters along the way. Remember, though, that he is young and untrained in combat. Let's look on page 23 of Runequest to find out something about the cost of weapons and about Rikur's chances of success in using them.

HOW COMBAT WORKS

THE ATTACK

In RUNEQUEST, an Adventurer has a certain probability of succeeding with an attack. If the player rolls the character's needed percentage or less on D100, the character has succeeded and managed to hit his opponent.

How much damage this hit does depends on the type of Weapon Damage and what kind of armor the target may be wearing to absorb the damage.

THE PARRY

The Adventurer also has the opportunity of parrying with shield or weapon. This chance is again rolled on D100 and, if the needed percentage to parry or less is rolled, the parry will block the attack, whether it was successful or not.

BASIC CHANCES

The Basic Chance for most attacks or parries (and most skills) is 5%.

A roll of 01-05 on D100 will always mean a successful attack or parry. Conversely, a roll of 96-00 on D100 will always mean an unsuccessful attempt at whatever is being attempted.



D100? Well, you can roll D100 by using two icosahedral (20-sided) dice. These are sometimes called percentile dice; the

twenty sides are labelled 0 through 9, with each numeral appearing twice. Use one die for the tens digit and the other for the ones digit. If you roll 00, call it 100. Or, of course, you can use the computer to generate a random integer in the range 1 to 100.

SPECIAL BASIC CHANCES

Certain weapons can be used with a Basic Chance higher than 5%, due to simplicity of handling and common use within the culture. The following list shows the Basic Chance to attack and parry with each of these weapons.

The table, on page 23 of Runequest, lists 22 weapons. Here are a few examples.

WEAPON	% CHANCE
Medium Shield*	10
Spear	10
Shortsword	15
Large Shield*	20
Dagger	25
Fist**	25
Kick**	25

*Parry only

**Attack only

So, let's see about outfitting Rikur with some weaponry. Page 28 is a table of WEAPONS, TRAINING COSTS, PRICES AND OTHER STATISTICS. Here are some samples.

WEAPON	STR/DEX	DAMAGE	PRICE
Spear	9 7	1D6+1	10
Shortsword	- -	1D6+1	10
Dagger	- -	1D6	15
Fist	- -	1D3	-
Kick	- -	1D6	-

EXPLANATION OF HEADINGS

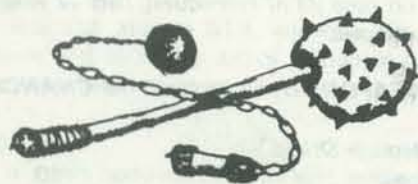
STR/DEX—The minimum necessary STR and DEX to be able to handle the weapon. A lack in DEX can be made up by an excess in STR, on a two-for-one basis. For instance, Rurik lacks one DEX Point of being able to use the Spear one-handed. Fortunately, he has a STR of 12, which is three over the nine needed. The extra point of DEX is made up with two points of Rurik's STR.

DAMAGE DONE—This is expressed as a Dice Roll (such as 2D6) plus, in many cases, an additional point or two of damage. Thus, the damage done by a Short Sword is found by rolling D6 and adding one to the result. The damage ranges between 2 and 7 points.

PRICE—This is the price of the weapon itself. It is a standard price and a better quality weapon could be offered at a higher price. Of course, "better quality" may just mean gaudier. . .

And, from page 30, some info on shields.

SIZE	STRENGTH	ABSORBS	PRICE
Medium	9+	12	10L
Large	12+	16	20



SIZE—Relative size of shield. A Small Shield is about 30-35 centimeters in diameter and held with one handgrip. It is called a buckler.

A Medium Shield is about 60 cm in diameter, equivalent to the ancient Greek Round shield, or the Medieval knight's "heater."

A Large Shield is the equivalent of the Foot Shields of the Greeks and Romans, or the Norman Kite Shield and the Viking Round.

STRENGTH—Strength necessary for a character to have to be able to use such a shield.
ABSORBS—The shield will absorb this number of points *per attack* before the user takes damage.
25%/50%/75%—Serves the same function as it does in the weapons chart. Again, any ability beyond 75% must be gained from Experience, not Training.
PRICE—The standard Price for a shield. Fancy ones cost more.

Rikur, who is not wealthy, decides to buy a shortsword, a dagger and a medium shield for a total price of 35 lunars. Of course, he already has built-in hands and feet, which cost him nothing.



KEEPING TRACK OF RIKUR

Now we need a way to keep track of Rikur. Here is what we know about him so far.

NAME	RIKUR			
STR	12	WEAPONS		
INT	16	TYPE	Attack %	Damage
POW	12	Shortsword	15	1D6+1
CON	16	Dagger	25	1D6
DEX	6	Fist	25	1D3
CHA	10	Kick	25	1D6
SIZ	12	SHIELD		
		SIZE	Absorbs	Parry %
		Medium	12	10

Rikur's record, of course, will grow much larger. In a game of Runequest, there will be many records! So, start thinking about how to build a Runequest data base management system. See you next time.

EPIC GAMES for Modest Computers



BY ARTHUR WELLS, JR.

Arthur lists his address as 1171 Cragmont Ave, Berkeley, CA, 94708. It seems he is a barrister who also has interests in computers, fantasy role-playing games, humor (see his letter in the Jan/Feb 1979 issue of RC) and much more. He requests that you fill his mailbox with replies about this article. He asks that the person who filled his mailbox with sand please discontinue correspondence. He asks that I stop this . . . -RZ

The article "Epic Computer Games," which appeared in the March-April 1978 issue of this magazine, is, as far as I am concerned, one of the best articles so far written on what games need to be. The article has some good concrete suggestions for implementing games and touches on a variety of topics, each of which needs much exploring. I also found it fitting that the article should appear in RC, which has always presented the pedestrian detail of programming with gusto and has usually tried for as much inventiveness as has been thus far possible in the recreational computer field.

It seems to me that the Epic Game is already well within reach and that within two years we could have some great games going. But I think this is going to require some attention to what the market makes available and to some of the realistic limitations on home computer use.

I come from a completely different background than the authors of other games articles. I learned BASIC at the Lawrence Hall of Science time-sharing system at Berkeley a few years ago, played all the games there, and for about a year have had my own home computer, which is an Apple II. I started with tape and now have a disc drive, and I have gone from 16K to 48K memory on board. I understand machine language programming, but have done little of it and I favor it only if speed or some other advantage presents itself as the *clear* choice. I have had exposure to other high level languages, so (I think) I grasp the different bases on which computer languages can be constructed.

© 1978 by Arthur Wells, Jr.



However, I have never written a program in FORTRAN or COBOL, nor have I ever had the opportunity to play with a big, really fast, powerful machine with 32 bit words and oodles and reams of memory. It is unlikely that I will ever get that opportunity, and I probably would be frozen into inaction if I were. In other words, I am just a hobbyist.

From this perspective, I have the following observations about Epic Games on home computers.

One BASIC. I have no doubt that we are going to have these games written in BASIC. Radio Shack, Commodore, Apple, Texas Instruments, and everyone else has already made a big commitment to BASIC. There are a lot of good books out to teach the language and the language is fundamentally easy to learn. The programs will obviously need an extended BASIC. What is needed most is *one common BASIC syntax* to run on all the different computers. The present situation where each vendor has a different dialect but any one can be made to say the same thing as any other is stupid and should soon become intolerable. One of the purposes of inventing the higher level languages was that programs not be chip- or machine-bound. Other languages have some useful functions but I believe we will have to contend with BASIC.

N-Dimensional Adventurers. One of the most exciting things about Epic-type games is the possibility of users defining their own player characteristics. It seems to me we already have the capacity to start developing complex personalities, even with limited memory, by using multi-dimensional matrices. Many of the advanced BASICs permit matrices of n dimensions, and, as I understand it, this means that a large number of characteristics can be related to a large number of other characteristics. So we can have characters with a dimension which moves them to tears by sorrow, or happiness, or relief, or frustration or anger, while simultaneously being sick, healthy, married, single, etc. Whether tearful or not, their anger can be simmering or violent—or anywhere in between.

It also seems to me that these personality matrices could be made to interact with situational or positional matrices (or both), so that the outcome is often not predictable. Obviously, the situational or positional (e.g. characteristics of a location) matrices can be complex or simple.

I think some intense work is going to have to be done on decision trees or matrix interactions, so that much more than simple branches are available to determine the action taken as a result of player/object/place interaction. The simple tree itself seems of little value for more than the "which-fork-in-the-road" decision. There must be many other

ways. For instance, most BASICs allow for at least a dozen nested loops, and I don't see why player/action can't be determined either by loop groups or a series of loops.

Modular Gaming. The notion that people, objects and places are defined by sets of characteristics makes for a modular set-up. This means that the same game or situation can become entirely different with different players and places. Assuming appropriate hardware (discussed later), each person/place/object could be a separate table/disc/tape. In this way, the game itself, i.e., the procedures available, could be kept short and simple (and perhaps in memory) but the game would vary greatly because of player/object characteristics. A game of football between the Raiders and the Rams would be vastly different than football between dragons and glorps (glorps are whatever you imagine), but the rules are the same.

Carry-A-Culture. This modular notion also suggests that the mere user can become quickly but completely involved. All this would seem to require is separate programs which ask a lot of questions, the answers to which create the matrix or table that defines the person/object. Within the limits imposed by the game to be played, the user can define himself or anything else as male/female, timid/aggressive, etc. He could choose passivism over aggression. And so on.

If this modular system can work, and if vendors will agree on a common syntax, then we can carry our small cultures and their inhabitants to friends' houses for interactive play. That is, if the necessary hardware is available, which is what we need to discuss next.

Hardware Support. Much of the hardware for really super games is already or soon will be available on microcomputers. As of now the following is available on at least some systems:

- Low resolution (40 X 40 grid) graphics in at least 15 colors.
- High-resolution graphics running up to 300 X 300 dot resolution in at least a half dozen colors.
- Hardware which permits you to get the computer to respond to between 32 and 64 different voice commands.
- Sound reproduction which will simulate five octaves of normal tones, all types of war or Starwars-like sounds, screeches, burps, bells and whistles, etc.
- Voice modules which are user-created, on call, can make sounds of struggle, grunts, sighs or regular speech. (Thus far the fidelity is poor.) On the Apple, the program was developed by Bob Bishop and Bill Depew of Softape and modified by Andy Hertzfeld, a graduate student at Cal.
- The Apple II can play chords of up to four notes with simple hardware.
- Speaker output from the microcomputer can go through your high fidelity/stereo equipment.
- There is presently equipment made by Mountain Hardware which will let your micro turn on and off lights,



alarms, etc., in your house. Thus your Epic Game can affect your whole apartment (. . . or cave?).

- By next year, Mountain Hardware says it will have biofeedback equipment which will work through the micros. This, theoretically, means control by skin response, heart beat or alpha waves.

Obviously, a mountain of work needs to be done both on hardware and software to bring these capabilities to many systems and at a decent price. The point I want to make here is that the home computer field is already fecund. Epic games could already be tending toward the ultimately desirable characteristic: outrageousness.

A Bottleneck. The biggest problem which seems to be current is how to access all this capability on the micro. Obviously, 4K is not enough; 16K is a minimum; 48K onboard better. But a really flexible and interactive system (player/player/machine) will not be realistic until we have disc storage with truly random access. I do not believe

anyone cares about micro-second responses. Five to ten second response would be adequate, especially if we had something in between to keep us entertained (such as a screen display). With disc storage, random access and decent on-board memory, we should be able to have fairly elaborate and interesting games.

Future Stuff. How about micro owners playing with each other? Modems will obviously be available and cheap, but phone lines cost money and prevent incoming calls. I cannot conceive a large computer sitting somewhere dedicated only to epic games. So it seems to me that the following are reasonable probabilities:

- Phone lines are used but an interrupt is built in to notify about incoming calls;
- People's micros are portable and can interface with each other at each other's homes; and
- People should be able to play at home without others; and micros have multiple keyboards so that two or

three players can all get into a home game. Further, the home set-up must be such that people can develop their own player/object/place matrix modules at leisure and at home, and then try them out. Disc is the only answer to this, and it will have to be true random access.

Summary. Let me finish this article with two minor observations. The first is that there definitely should be different levels of play for different age/maturity groups. In my experience, children 8-12 are going to feel a lot less restrained about suggesting or trying changes which adults would either not think of or would fail to suggest for fear of criticism. The future of the truly inventive and creative game lies in the hands of those untroubled by peer criticism.

Lastly, every game incorporates moral judgments. Whether knights protect maidens or kill them can involve moral judgments. So can definitions of "winning." The importance we give these judgments can be separated from the judgments. The key thing is that the program should not unduly punish failure or an obviously morally wrong decision.

For instance, Apple puts out a game called "Noah's Ark." It is a "hangman" game, except that the words to guess are all biblical. If you miss a letter the ark starts to sink. If you fail to guess the word in the requisite number of guesses, the ark sinks. Thereafter, a sea of blue appears on the screen and an upside down human body "glub-glub-glubs" to the bottom. If you win you get a printed "You live . . . for now." Peter Rowe of LHS has lucidly observed that the reward for failure exceeds the reward for success. One can see that this tendency definitely must be curbed so that rewards for success—either visually or vocally or in terms of game "currency"—exceed those of failure. This may require overcoming a natural tendency to make failure visually and aurally more interesting than success.

Dozens of other ideas probably come to your mind at this time. However, this article must end somewhere and this seems like a good point. I insist on hearing responses, and you should all feel guilty if you do not inundate my mailbox with replies. □

UNIVERSE

An Immodest Proposal



BY LES LAZAR

Think big. Cosmically, if you can. That's the only way to approach Les LaZar's Universe, the computer game to end all games. Or to begin all games. The one that will reduce Super Star Trek to a mere warm-up exercise.

Les, who by his own description is slightly nuts, wants to create a structure to encompass "all games in one—all that ever were and all that ever shall be..." Amazingly, the article that follows makes it seem possible. Insanely ambitious, yes. But possible.

For it to happen, however, visionary Les needs a little help from his friends. So read on, and if you have any ideas for aiding and abetting this grand scheme, write to him at 14701 Arminia Street, #J, Panorama City, CA 91402. You, too, can conquer the universe. —LB

The following pages propose an elaborate computer game called Universe. It's the kind of game I would like to play, since it contains elements of the most enjoyable and challenging games I've experienced (specifically, Super Star Trek and Adventure). But it goes beyond them in an attempt to provide an unlimited structure upon which to build a basic game and any number of additional games. If the proposed structure isn't appropriate to these goals, I trust one of you who is a better programmer will suggest an alternative.

My primary problem with existing games is boredom. While Adventure kept me busy for a few weeks, it was essentially a puzzle, and once solved, was of little additional interest. And though Super Star Trek is different each time, even it seems the same after a while: no new challenges, no new experiences. Zapping Klingons gets old after the 300th voyage.

The game I am trying to define is intended to give the player the initial challenge of Super Star Trek or Adventure—but keep it up, indefinitely, if possible. How can it be done? Well, what I am proposing is, if you will, all games in one—all the games that ever were and all that ever shall be . . .

Copyright © 1978 by Les LaZar

If you've gotten this far, you may think I'm nuts—or you know me, and know I'm nuts—and just want to see what I'm getting into this time. Well, read on, and you'll find out.

First, let me say that I realize what I'm proposing can't be done all at once or by a single person. That's why I submitted this article to *Recreational Computing*. (They are as crazy as I am.) I am soliciting input and help. If successful, we who create Universe will be able to play a very challenging game, and our accomplishment will be recognized by all players who subsequently encounter the program.

THE ROOT OF THE MATTER

Universe is, in essence, a framework for any game with some ideas of my own for an initial set of games to build on the framework. You may have your own ideas on both the framework and the games. To use programmers' jargon for a moment, Universe consists of a *root* and a set of *overlays*, which are loaded into the system as needed. The root will provide the basic utilities that, when combined with one or more overlays, constitutes a playable game. This structure is similar to Adventure, except that the root segment will be kept totally general; any data or program segments that are specific to a particular game will be kept in the overlays.

With your help, we can design a root that will be general enough that any conceivable game can simply be *plugged in*. The root will provide as complete a set of game-oriented utilities as possible, consistent with the requirements of memory space and generality.

The root itself may employ overlays for general functions that are not used enough to justify residency, but are popular enough to be part of the basic utility set. Any function not included in the root must be provided for in each overlay that uses it (although overlays may invoke other overlays from a non-resident library for the semi-general functions and databases). As you can see, the organization is getting complex fast.



The overlays are what turn the root into a game. A simple game may require only one overlay, while a complex scenario may use several. The partitioning of the root and the overlays must be done carefully in order to create elaborate games that are interactive and still can run in limited system memory. (I am assuming that at least two, single-density, floppy disks with 512Kb are available for on-line mass storage).

A classic conflict is readily apparent. If the root is very large (and provides lots of functions and databases), the overlay space will be limited and, for complex games (which is what Universe is all about), many disk accesses will be required, slowing the response time. On the other hand, if the root is chopped to the bare bones, then common services may have to be duplicated in the overlays, thus making them larger than necessary; chewing up valuable memory and mass storage space. Some hard decisions will have to be made; the wrong choices could doom the project. Anyone experienced in this area is requested to submit a solution. If there is some way to make a soft (i.e., changeable) division between root-resident utilities, root-overlaid utilities, and functions necessarily provided by the overlays, I would like to hear about it. I fear that once the root functions are defined, the die will be cast, and Universe will fail or succeed on the basis of that early decision. It might be helpful to think of the root of Universe as an interpretive game language, in much the same way that BASIC and APL are interpretive languages, with the overlays being programs (and data) that are written in, and run under control of, that language.

But enough of this dry programming analysis. Let's get on to the fun part of this project.

THE BEGINNING GAME

I'm into science fiction, as I suspect many of you are, and I enjoy playing games like Super Star Trek, which put me into a space scenario/environment.

As I envision it, the initial game of Universe will have the entire Milky Way as its playing field. The player (or players) will have at their disposal a spacecraft, crew, material assets (such as trade goods,

weapons, money, etc.) and, rather than a single scenario ("Kill Klingons"), a choice of possible goals. The choices will expand as new games are added to the library by fanatics like us, who not only play, but also program.

The primary game will go something like this:

The player has just come of age. The government of the federation of planets, in order to encourage development of space, will loan the player funds (federation credits) to purchase and outfit a ship, raise a crew, and provide initial working capital. The player, given an initial amount of starting capital, must decide what type of ship to buy, how to equip and crew it, and how much cash to save for expenses. The game will include a written manual to give the player some direction for his or her choices.

There are three possible goals of the game: to accumulate money, to gain "experience" (quantified in terms of points based on the player's adventures), and to have fun exploring the galaxy. The economic goal will tend to pervade the game, as it does real life. The mortgage payments on the ship must be made, the crew payroll must be met, and expenses must be covered. This means that periodically the player will have to do something socially useful, or at least profitable, like gambling—and winning—in order to get the money to support his or her other activities.

What can the player do to meet his or her financial obligations? Well, the player has major capital assets: the ship, crew, and working cash. And the player can earn money by undertaking missions. Every spaceport will have a bulletin board where individuals, organizations, and government agencies can list available missions. What kinds of missions might one find?

Exploration-survey. Since the volume of explored space is very small, the government's Department of the Exterior has survey assignments to previously unvisited areas. Even if you find a vacuum, the DEX fee will cover your expenses and provide you with a small profit to gamble away; then, next day, off again. However, if you find something interesting, like a star with planets, or a black hole, there will be a nice bonus. And, in the unlikely,



but possible, instance that you find something really interesting or valuable, like an inhabited planet or precious minerals or technological artifacts left by the long extinct Slavers (*dum-de-dum-dum*) you could get really rich—if the inhabited planet isn't actively hostile or pirates don't take your cargo of platinum or the Slaver stasis box doesn't contain a billion-year-old cheeseburger. Such are the risks of space travel.

Another class of adventures awaits you if you discover a new planet. While many aspects of a planet's nature can be determined from orbit, the real excitement (the large profits and risks) are to be found on a surface survey. Each planet will be different—size, terrain, flora, fauna, minerals, inhabitants, customs, you name it—and the computer will create it. Once you reach the ground, you will begin the search for knowledge and profit.

You may explore in vehicles (if you brought them with you) or on foot, in vacuum armour or shirtsleeves. Needless to say, surprises lie behind every rock. Perhaps even the rock itself presents an opportunity for danger or riches. Don't let the local bacteria or the dire beast get you, but don't overlook the binqi bush (much needed back on Regulus III to cure the creeping scourge) or the gell crystals (used in very dense computer circuitry). Even if you find nothing of monetary value, merely surviving in an alien environment and getting away safely will earn you experience which will come in handy on subsequent adventures. The greater your experience point total in a variety of categories, the greater your chances of escaping subsequent dangers and being successful in your missions.

The Free Trader. There is a constant commerce between the developed planets as well as the more primitive colonies. The spaceport provides up-to-the-minute quotes on buying and selling prices of various common and exotic commodities on the principal planets of the federation. Typically, the Free Trader will buy items that are cheap locally and transport them to a market where a high price is being paid. This may be a principal planet, whose prices are quoted on the spaceport exchange or a minor settlement whose needs you can only guess. In addition to direct exchange of goods for federation money, some planets may run a barter

market. You must trade goods for other goods: they neither accept nor pay in monetary exchange. This entails considerable risk, but the potential for profit is high.

A typical trading trip might progress as follows: you buy a load of machine tools at the industrial colony in the Centauri system; these are sold on Sirius V, and a load of tractors, combines, threshers, and porno films is acquired; these are sold on the agricultural world of NGC2385 for a combination of federation credits, edible produce, and seed stock (both plant and animal).

As you get farther away from the developed worlds, federation credits mean less, and tangible goods are the medium of exchange. In the asteroid mining operation of the Cygnus system, fresh fruit and vegetables are worth their weight in refractory metals, so that's what you get. The Cygnus miners also brew a potent platinum liqueur, and you trade your last reel of "Deep Venturi" for a case of it.

Now starting to swing back toward civilization (?), you make a stop on the recently discovered tropical planet, Amazonia. At the moment, the only inhabitants are a few settlers who eke out survival by farming, hunting, and trapping. They are of the Au Natural sect and need nothing you have. But one of the children is sick, and your ship's physician treats him (you did include a medical doctor on your crew, didn't you?). In payment, you are offered some smelly old mithra pelts. You accept the furs, knowing they are used to make the best reverse osmosis desalination membranes, and thus are the most valuable items in your cargo—worth half a year's ship expenses.

The Charter Skipper. The player can also earn money through a charter service, transporting cargoes and passengers for which a scheduled commercial ship is inappropriate. The cargo may be dangerous or valuable; the passengers may be in a hurry; there may be a longspacemen's strike in progress or, as is usually the case, there is simply no commercial way to get there, and a charter ship (yours) is needed.

These missions will pay a fixed fee for performance on schedule. Various monetary penalties will be assessed if you do not meet your delivery deadline. Pirates



may attempt to hijack your cargo. You may have been hired by smugglers trying to avoid planetary customs officials. The list goes on and on, limited only by your imagination and willingness to program the various scenarios.

LEVEL OF TECHNOLOGY

All these scenarios, of course, require a high level of space technology. You can create any type of technology you wish to program; the capabilities described here are only starting points.

Space travel. An interstellar civilization 100 light years across would seem to require faster-than-light (FTL) travel. For "normal" travel, photon-drive (from direct conversion of matter to energy) would be sufficient, but FTL speeds would require "hyperspace jumps"; these would transport your ship several light years in zero time. (I have worked out some of the problems with hyperspace jumping in a longer paper.)

Communication. I think my civilization requires instantaneous communication, primarily for collecting ship mortgage payments while traveling and for transmitting commodity prices (free trader scenario). However, this type of communication is not easy. It requires the generation of beams of tachyons by giant particle accelerators. These "transmitters" are 20 miles long and require enormous amounts of energy. Only principal planets and the largest of orbital and traveling colonies have the technology or the space for these machines. The receivers are million-gallon vats of Gatorade, which flash as the tachyon beam is generated.

Depending on the energy input to the transmitter, the resulting flash in the receiver is at a particular frequency (color), and thus the beam can be tuned and modulated. So, while there is fast communication between the principal population centers, the less developed planets and spacecraft must rely on radio or courier ships for their mail and news.

IMPLEMENTING THE DREAM

At a minimum, I am assuming Universe will have a system consisting of an LSI-11 with dual single-density floppy disks, 28K memory, EIS/FIS (floating point arithmetic), line clock, and video

terminal. I picked the LSI-11 because it is the machine I have access to and is an excellent, inexpensive minicomputer.

The game should be able to utilize a line printer for certain types of output and a larger mass storage device. Extensions to the game may require graphic output and analog input, but these options would not be required for a basic version. If the software is designed correctly, Universe should be easily transportable to other CPUs.

What language will it be written in? Here is a question that will generate a lot of discussion. The decision on implementation language must be made early in the project and will affect how many of you eventually participate in the creation of the game program (root and overlays). In order to fulfill the requirements of a complex, interactive game, the final, executable code must be compact. Since the root and the overlays will be extensive, the space they occupy in memory or mass storage must be minimized. Also, the code must execute fast because of the interactive nature of the game and its computation requirements.

The ability to overlay from mass storage is essential. Both the root and the overlays must be able to call data and additional overlays from mass storage in order to support the various scenarios and environments of the game. A flexible and dynamically allocatable memory usage technique will be required.

The source code should be readable by persons other than the original author. Since the creation and extension of Universe will be a long-term effort, a widely available and understood language is a must. A language that supports "structured" programming constructs is desirable.

With these criteria in mind, I will now give you my opinions on the various languages available for the hardware mentioned above.

ASSEMBLY (MACRO II). While MACRO will fill the speed, size, and flexibility requirements, its readability by other than the author is less than optimal. MACRO understandability is extremely dependent on the author's structuring and commenting skills.

BASIC II. Forget it! While BASIC is easy for small programs, its limitations cause it to get unwieldy in large configurations. BASIC also lacks the necessary flexibility in using programs and data from mass storage. BASIC is a pig when it comes to speed and the size of the code modules. Also, support for real-time programming (line-clock-interrupt-driven) is missing in BASIC.

FORTRAN. Here is a good candidate, if properly documented. The major games I have discussed—SST and Adventure—were written in FORTRAN (with MACRO subroutines when needed). FORTRAN meets most of the criteria described above and is already used by many of you. Its limitations for the game (string manipulation, data structures) must be overcome by clever programming. I would prefer to concentrate the clever programming on implementing new scenarios rather than on the basic structure.

PASCAL. A relatively recent language, PASCAL is rapidly gaining in popularity. Unfortunately, I am not as familiar with this language as those already discussed. It certainly seems to have FORTRAN beat as far as ease of game implementation goes, since it has a wide variety of complex data structures available. Another advantage is that it does not require RT-11 (I am referring to the UCSD PASCAL system which includes its own operating system, editor, utilities, etc.).

The avoidance of an expensive operating system could enlarge the potential pool of contributors and players. Additionally, UCSD PASCAL is not limited to LSI-11 (or PDP-11), thus increasing further the number of interested parties. Since I have not programmed anything in PASCAL, I must depend on those of you who have to evaluate it as an implementation language.

NITTY-GRITTY CONSIDERATIONS

As I have stated before, Universe will consist of a root, containing general utilities and a database, and a set of overlays which actually implement the various scenarios and environments. The root would be a sort of game language, interpreting portions of the overlays. The root

is not a language in the familiar sense, however; the overlays would contain code written in the implementation language. This code would execute independently of the root, calling root functions as needed. The root would also interface to various data structures of the overlays, such as decision tables.

The root would have the following functions:

1. Keeping track of global variables, including game time and date, player status (money, experience), and player location and motion.
2. Accepting commands from the player and, in conjunction with the "current command vocabulary" (one of the data structures passed to the root by the currently controlling overlay), passing a command code with appropriate parameters to the current control program.
3. Displaying the current status to the player in conjunction with display routines in the current control program.
4. If the latest status change requires it, bringing in a new database or control overlay from mass storage.
5. Allowing the current control program to update the permanent database as necessary.

In addition to the above command parsing, display, read-and-write functions, the root should also provide a selection of other services to the control programs that will minimize their size without inordinately causing the size of the root to grow. Such services might include:

1. Random number generation, in a specific range, with specific distribution characteristics (linear, Gaussian, user specified function).
2. Using the random number generators and decision tables, supplied by the overlay, to return a set of results to the overlay.
3. Some mechanism for making selections based on conditional probabilities, something table-driven that could use local and global variables as the parameters.

The root would also contain a common database (possibly overlaid) that would provide a common command vocabulary and a common set of output messages and displays.

The components of an overlay are: local command vocabulary, the control program, permanent database (including local variables), decision tables with probabilities and results, and local output message list and displays.

A particular scenario (game) may be so complex as to require several overlays. These additional overlays may contain one or more of the above parts as the need arises. Obviously, a flexible organization is required to provide program responsiveness and still remain within the constraints of main memory.

AS FOR EDUCATION...

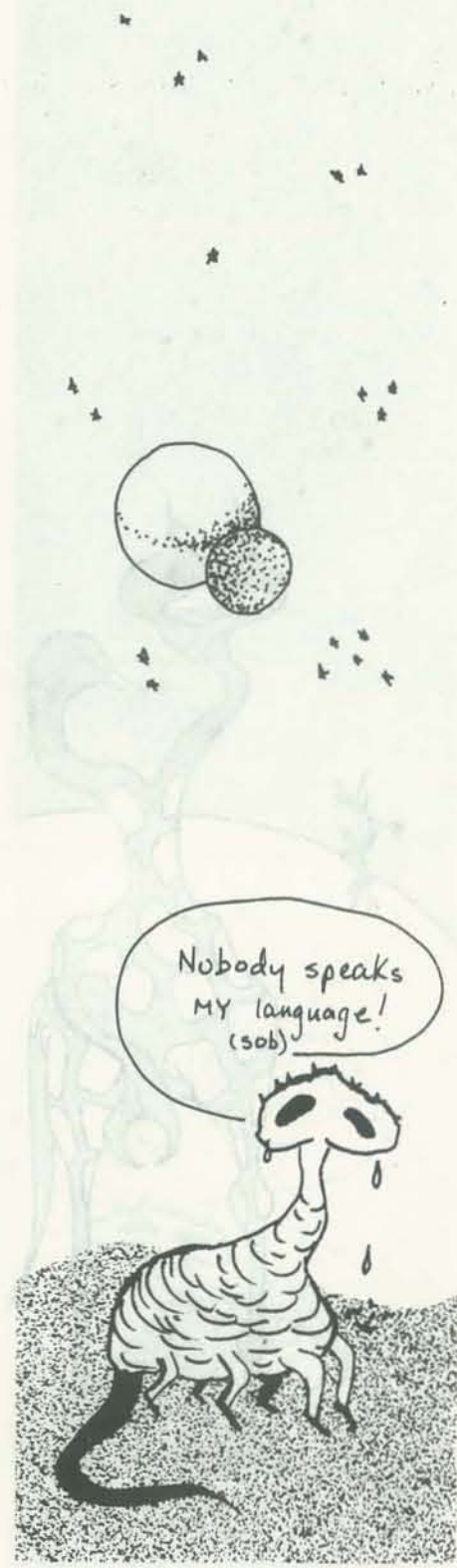
I hope the scenarios of Universe will be programmed to be as realistic—and educational—as possible. For instance, in the Survey-Exploration game, the player will learn about local astronomy; the names, locations, and characteristics of the nearby stars will all be in the permanent database. In an unexplored area, the decision tables that generate the stars and planets will reflect current knowledge of star type distribution and likely planetary characteristics.

Other scenarios may teach geology, sociology, linguistics, problem solving, and resource allocation—as well as economics, anthropology, psychology, and how to play craps.

THE LONG VIEW

Imagine, if you will, the structure of Universe becoming widespread, with hundreds, or even thousands, of players. There could be a newsletter announcing new scenarios and bug fixes and annual conventions with multi-player interactive tournaments. (Will PASCAL support multiple terminals?) Once an active player base is established, it is not inconceivable that new additions to the scenario (or database) library would be *paid* for by eager players hungry for new adventures.

Well, there it is. A game to fill the universe! I hope it interests you enough to get involved. I can't do it alone. Please contact me with your ideas and words of support. □





Easy POKEing with Applesoft™ BASIC

BY CHUCK CARPENTER

The climate in Carrollton, Texas, must be perfect for Apple users. Chuck is sending us a steady stream of Apple articles and promises more. Perhaps he will soon have enough for a book. For the title he might choose: "A carpenter from Carrollton discovers new uses for the Apple." POKE along little dogie... -RZ

POKEing machine-language programs in from BASIC can be clumsy. Especially if your program is longer than a dozen or so bytes. The utility programs in Figures 1 and 2 make the task as easy (well almost) as typing RUN.

The first program example (Figure 1) is self-contained. Line 140 has the starting location of the program. In this example, it is \$360 (864 decimal). The \$ indicates a HEX (hexadecimal) number. Line 160 has the machine language program entered as a string. String parsing (resolving into component parts) is accomplished by the subroutine in lines 190 through 270.

Starting the program automatically is accomplished in line 180. A CALL to decimal location 864 (\$360) starts the program for you. In this example, the

program is my printer driver. I use the TTL output from the game paddle annunciator (ANO), pin 15. My Telpar thermal printer has the appropriate input interface. This way, I don't use any other hardware to print listings and programs. You can use it to slow list your programs, too. As is, the routine runs at 110 baud. Putting a \$4D in location \$394 makes it run at 300 baud. (See Listing for the complete program.) Turn the program on with a 360G or CALL 864. Turn it off with PR#0 or RESET. The examples in this article were printed using this routine.

Figure 1. POKE Program with Printer Driver Routine.

```
100 REM *****
105 REM * ASSEMBLY LANGUAGE POKE *
110 REM * ROUTINE -- APPLESOFT FP *
115 REM * BASIC *
120 REM * BY:BOB SANDER-CEDERLOF *
125 REM * FP MODIFICATIONS BY: *
130 REM * CHUCK CARPENTER *
135 REM *****
140 LET LOC = 864
160 LET H$ = "A9698536A90385376084354828850368C98
ID00CA98A208503A95820A8FC998DA4354CF8FDA00B1848B00
5AD58C89003AD59C0A9D348A9204A90FD68E901D0F5686A88D
0E360"
170 GOSUB 190
180 CALL 864: PRINT : PRINT : PRINT "DONE": END
190 FOR I = 1 TO LEN(H$) STEP 2
200 LET H1 = ASC ( MID$( H$, I, 1) ) - 48
210 IF H1 > 9 THEN H1 = H1 - 7
220 LET H2 = ASC ( MID$( H$, I + 1, 1) ) - 48
230 IF H2 > 9 THEN H2 = H2 - 7
240 POKE LOC, H1 * 16 + H2
250 LET LOC = LOC + 1
260 NEXT I
270 RETURN
```

Figure 2. POKE Program with String Input Capability.

```
140 HOME : PRINT : INPUT "INPUT START LOCATION - ";LOC
145 PRINT : INPUT "INPUT GO LOCATION - ";GO: PRINT
150 INPUT "INPUT 2 DIGIT HEX VALUE - ";A$
155 IF A$ = "END" THEN GOTO 170
160 LET H$ = H$ + A$
165 LET C = C + 1: GOTO 150
170 GOSUB 190
175 PRINT : PRINT C: PRINT : PRINT H$: END
180 CALL GO: PRINT : PRINT : PRINT "DONE": END
190 FOR I = 1 TO LEN(H$) STEP 2
200 LET H1 = ASC ( MID$( H$, I, 1) ) - 48
210 IF H1 > 9 THEN H1 = H1 - 7
220 LET H2 = ASC ( MID$( H$, I + 1, 1) ) - 48
230 IF H2 > 9 THEN H2 = H2 - 7
240 POKE LOC, H1 * 16 + H2
250 LET LOC = LOC + 1
260 NEXT I
270 RETURN
```

Apple mini-assembler listing of the printer driver program.

```
*360LL
0360- A9 69 LDA ##69
0362- 85 36 STA #36
0364- A9 03 LDA ##03
0366- 85 37 STA #37
0368- 60 RTS
0369- 84 35 STY #35
036B- 48 PHA
036C- 20 85 03 JSR #0385
036F- 68 PLA
0370- C9 8D CMP ##8D
0372- D0 0C BNE #0380
0374- A9 8A LDA ##8A
0376- 20 85 03 JSR #0385
0379- A9 58 LDA ##58
037B- 20 A8 FC JSR #FCA8
037E- A9 8D LDA ##8D
0380- A4 35 LDY #35
0382- 4C F0 FD JMP #FDF0
0385- A0 0B LDY ##0B
0387- 18 CLC
0388- 48 PHA
0389- B0 05 BCS #0390
038B- AD 58 C0 LDA #C058
038E- 90 03 BCC #0393
0390- AD 59 C0 LDA #C059
0393- A9 D3 LDA ##D3
0395- 48 PHA
0396- A9 20 LDA ##20
0398- 4A LSR
0399- 90 FD BCC #0398
039B- 68 PLA
039C- E9 01 SBC ##01
039E- D0 F5 BNE #0395
03A0- 68 PLA
03A1- 6A ROR
03A2- 88 DEY
03A3- D0 E3 BNE #0388
03A5- 60 RTS
03A6- FF ???
03A7- FF ???
*
```

A value of \$20 in location \$394 provides a nice slow listing speed. Compare this program with the one in the Apple II manual on pages 119-120.

Figure 3. Example Input and Run of the Figure 2 Program.

```
INPUT START LOCATION - 768
INPUT GO LOCATION - 768
INPUT 2 DIGIT HEX VALUE - EA
INPUT 2 DIGIT HEX VALUE - 20
INPUT 2 DIGIT HEX VALUE - E4
INPUT 2 DIGIT HEX VALUE - FB
INPUT 2 DIGIT HEX VALUE - 60
INPUT 2 DIGIT HEX VALUE - 00
INPUT 2 DIGIT HEX VALUE - END
6
E820E4FB6000
IONE
CALL-151 Calls the monitor without using RESET.
*300L
0300- EA NOP
0301- 20 E4 FB JSR #FBE4 Rings the bell. (Control G.)
0304- 60 RTS
0305- 00 BRK
```

Programming POKES all in one string is fine for programs you use frequently and as is. And, you could combine several routines by putting in multiple start locations and HEX strings. But, if you want to key in experimental routines, retyping the string, H\$, each time gets old very soon.

The second example (Figure 2) solves the problem for you. Each value needed by the program is handled as a separate input statement. Line 140 asks you for the decimal value of the starting program address. Line 145 wants the value for the turn-on point of your program. (Values in line 140 and 145 may not always be the same.) Pairs of HEX values are entered through the input loop in lines 150 through 165. Line 150 accepts the HEX pair as a string, A\$. Ending the input string is done by entering END (clever, huh!). The program then jumps to line 170 where it runs to the end.

Stringing the H\$ string together (concatenating) is done in line 160. The first time through H\$ has no value. When the first A\$ is input, H\$ gets that value in line 160. Each time you enter a value at line 150, it is tacked onto the end of H\$ in line 160. When END is entered, the program branches past the other input lines. Your entries are POKE'd in by the subroutine starting at line 190 and RUN by the CALL in line 180. If you don't want it to run, leave out the CALL. A counter is included in line 165 and printed out in line 175 as the number

of HEX pairs entered. Also printed out in line 175 are the contents of your input string H\$.

Figure 3 is an example of the inputs, a run of the program, and results from the Apple's monitor. The listing from Apple's mini-assembler shows the results of POKEing in the H\$ string.

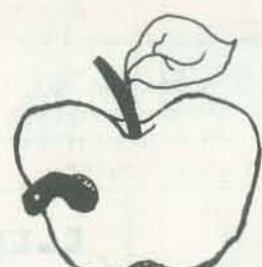
This POKE program was originally written in Apple integer BASIC by Bob Sander-Cederlof. About a year ago, when our local Apple Users Group first organized, Bob provided us with a number of helpful utility routines. Figure 4 is an integer BASIC version (for comparison) and is presented here with Bob's permission.

You may be curious about the values subtracted for obtaining H1 and H2. This has to do with the way Apple integer and Apple-soft BASICs handle the keyboard strobe bit (\$C000). Apple integer leaves it on the ASCII value and Applesoft does not. Other lines in the parsing-POKEing routine adjust the ASCII values of the HEX string to put them in the 0 to 15 range. (There are gaps between the decimal numbers and the alpha characters used to represent the hexadecimal numbers.)

Applesoft BASIC routines included with this article can also be used (with appropriate modifications) by PET, OSI and TRS 80 Level II owners. I hope you find them useful, too. □

Figure 4. Integer BASIC Version.

```
1050 LCT=2048
1060 DIM H$(32)
1070 H$="0C95F948201E0068C96DD00CA98A201E"
1080 GOSUB 1210
1090 H$="08A95820A8FC998DA4354CF8FDA00B1848B00
1110 GOSUB 1210
1120 H$="1848B005AD58C09003AD59C0A9D348A9"
1130 GOSUB 1210
1140 H$="204A90FD68E901D0F5686A88D0E360A9"
1145 GOSUB 1210
1150 H$="008D96A9A9088D97A960A9F08D96A9A9"
1170 GOSUB 1210
1180 H$="FD8D97A96000"
1190 GOSUB 1210
1200 END
1210 FOR I=1 TO LEN(H$) STEP 2
1220 H1=ASC(H$(I))-176
1230 IF H1>9 THEN H1=H1-7
1240 H2=ASC(H$(I+1))-176
1250 IF H2>9 THEN H2=H2-7
1260 POKE LCT,H1*16+H2
1270 LCT=LCT+1
1280 NEXT I: RETURN
```



those uppity

A wisecracking calculator stealing the show? Well, it had to happen sometime—and it did, last November. That's when the *Stanford Daily's* two cartoonists—simultaneously and unknown to each other—began featuring calculators in their strips. Either the artists were connected to the same muse, or it was an idea whose time had come.

Both Chris Juricich, creator of "J. Moist and Co." (on this page), and Gil Morales of "Dupie" (facing page) say they

were inspired by the talking computers in such movies as *2001* and *Demon Seed*. Why not, they reasoned, invent a calculator with a mind of its own?

Juricich, who received his bachelor's degree in political science from San Jose State University, says he's wanted to be a cartoonist since he was a kid, "but I kept finding excuses for not getting involved with art." A year ago he began jotting down ideas, carrying a notebook with him everywhere.

J. MOIST AND COMPANY

By Chris Juricich



© 1978 C. Juricich



© 1978 C. Juricich

calculators

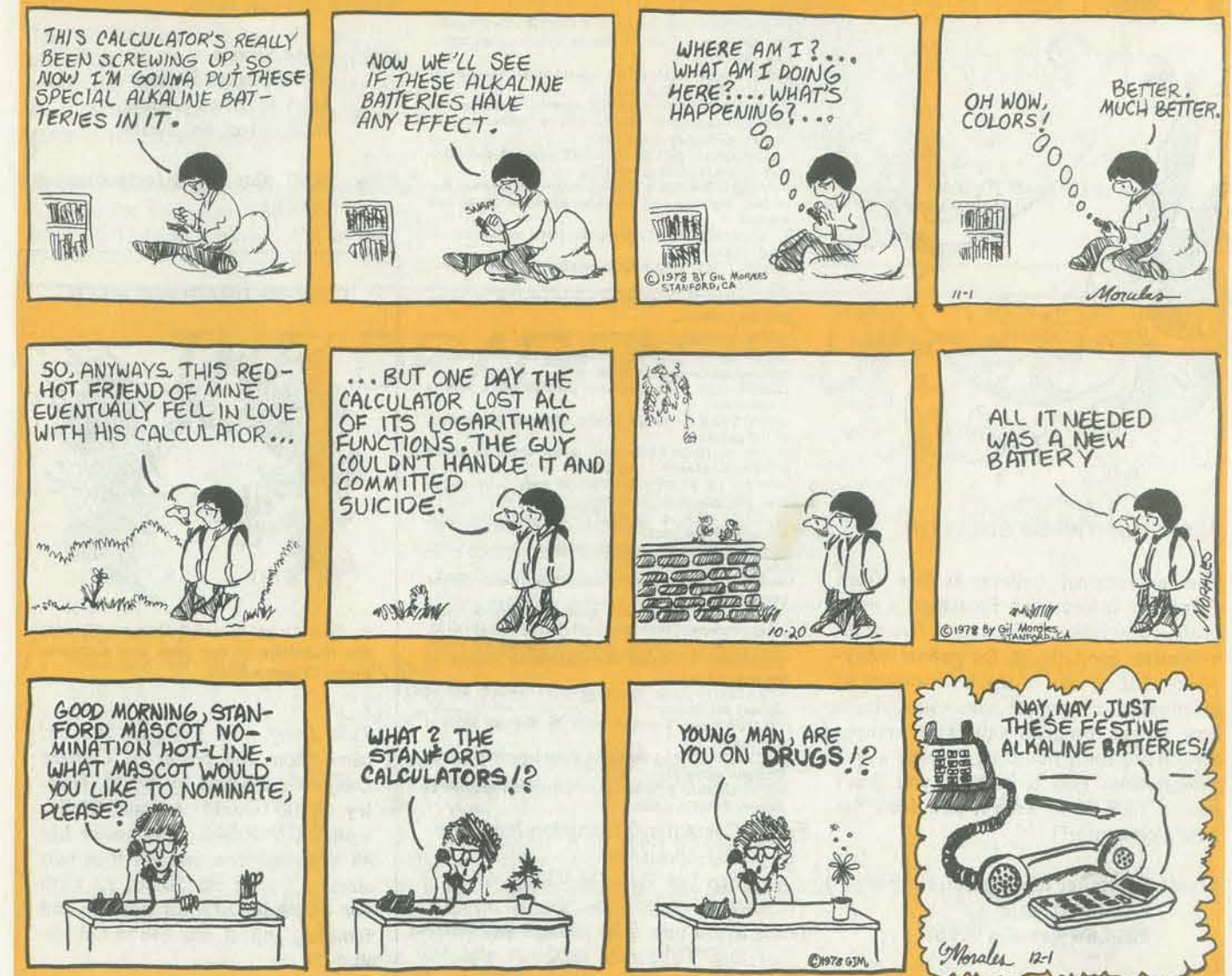
"I live with three other people, and many of my best lines come from them."

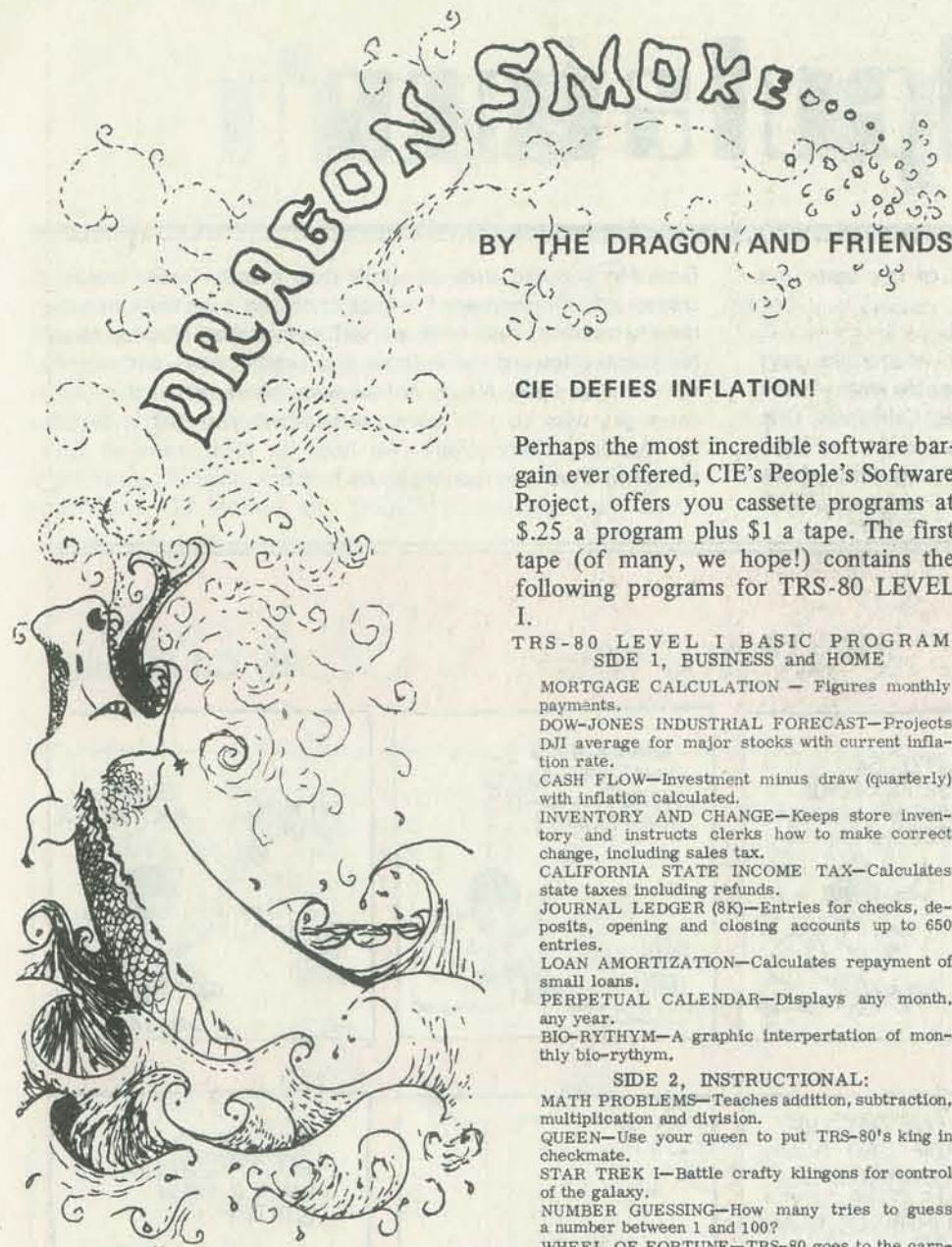
Morales, a Stanford sophomore majoring in economics, says cartooning is a great way to avoid studying. He learned that back when he was in high school in Whittier, California. One day, after the school newspaper's regular cartoonist had been expelled, Gil volunteered to take over. "I couldn't draw, and that first cartoon was *terrible*. I've definitely improved."

Both Morales and Juricich admit their debt to Garry Trudeau, creator of "Doonesbury," whose style and tone they imitated heavily at first. And both are well aware that Trudeau began his journey toward the Pulitzer in a similar way—with a strip for the *Yale Daily News*. A few years hence, when the syndicates get wise to this talent, remember: you saw it first in *Recreational Computing*! (We hope to print more of Gil's and Chris's work in coming issues.)—LB

DUPIE

By Gil Morales





BY THE DRAGON AND FRIENDS

CIE DEFIES INFLATION!

Perhaps the most incredible software bargain ever offered, CIE's People's Software Project, offers you cassette programs at \$.25 a program plus \$1 a tape. The first tape (of many, we hope!) contains the following programs for TRS-80 LEVEL I.

TRS-80 LEVEL I BASIC PROGRAMS SIDE 1, BUSINESS and HOME

MORTGAGE CALCULATION—Figures monthly payments.
DOW-JONES INDUSTRIAL FORECAST—Projects DJI average for major stocks with current inflation rate.
CASH FLOW—Investment minus draw (quarterly) with inflation calculated.
INVENTORY AND CHANGE—Keeps store inventory and instructs clerks how to make correct change, including sales tax.
CALIFORNIA STATE INCOME TAX—Calculates state taxes including refunds.
JOURNAL LEDGER (8K)—Entries for checks, deposits, opening and closing accounts up to 650 entries.
LOAN AMORTIZATION—Calculates repayment of small loans.
PERPETUAL CALENDAR—Displays any month, any year.
BIO-RHYTHM—A graphic interpretation of monthly bio-rhythm.

SIDE 2, INSTRUCTIONAL:

MATH PROBLEMS—Teaches addition, subtraction, multiplication and division.
QUEEN—Use your queen to put TRS-80's king in checkmate.
STAR TREK I—Battle crafty klingons for control of the galaxy.
NUMBER GUESSING—How many tries to guess a number between 1 and 100?
WHEEL OF FORTUNE—TRS-80 goes to the carnival with dazzling graphics.
WORLD WAR II BOMBER—Fly with the aces of the glorious past.
ROCK, SCISSORS, PAPER—Modern version of an old favorite.
SEEK—A challenging game using basic planar geometry.
STAR TREK III (6K)—A good space game using galaxy maps, sensors, and navigational skill.
RED BARON—Graphic flight to get the old kraut.
MINI-TREK—Compact but challenging version of the popular game.
STRATEGY—Keep track of your armed forces against the enemy.
PILOT—Above average skill is needed to land this plane.
BATTLESHIP—An excellent search game using coordinate geometry.
ON A SNOWY EVENING....—Delightful graphic of Robert Frost's poem.

From: Computer Information Exchange
 P.O. Box 158
 San Luis Rey, CA 92068

Price: \$7.50 plus \$.50 postage and handling. California residents add 6% sales tax.

A NON-VIOLENT SPACE GAME

Available soon, a non-violent space game from the World Future Society. We haven't seen it or played it yet, so we will simply pass on what they say about it.

Most games depicting humanity's ventures into space involve exploitation of other planets and warfare among the players for domination of space. Space Future is based on a cooperative approach. Players individually venture into space on peaceful missions and assist each other in settling and developing other planets. The game equipment consists of a large playing surface, a map of outer space with several planets along with resource tokens and several decks of playing cards containing directives and information. Ages: 10 and up.

Hmmm . . . Perhaps the Don Quixote Starship will soar again.

From: Book Service
 World Future Society
 4916 St. Elmo Avenue
 Washington, DC 20014

Price: \$9.50 plus \$1.50 for postage and handling.

CHINESE DRAGON POSTER



2. *Tien Lung*, Celestial Dragon. Guards the mansions of the gods and supports them. Chases a pearl.

Tien Lung, The Celestial Dragon, came from the poster of *Chinese Dragons* published by W. M. Hawley, 8200 Gould Avenue, Hollywood, CA 90046. This poster has 41 dragons. You can get it in two sizes: 17 x 22 for \$0.50; 22 x 30 for \$1. Add 20% for postage and handling and, if you live in California, 6% tax.

A NEW FRP MAGAZINE

Last issue we told you about a new magazine, *Sorcerer's Apprentice*, devoted to fantasy role-playing games. *Sorcerer's Apprentice* comes from the folks at Flying Buffalo in Scottsdale, Arizona. Since that report, another FRP magazine, *Different Worlds*, has made its debut.

Different Worlds is produced by The Chaosium of Albany, California, the same group that brought you *Runequest*, *Authentic Thaumaturgy* (see "Dragon-smoke," Nov-Dec 1978), and other magical adventures. *Different Worlds* is billed as "a new magazine of game role-playing . . . an authoritative journal that no dedicated gamemaster should do without . . . featuring articles on Dungeons and Dragons, Tunnels and Trolls, Chivalry and Sorcery, Traveller, Runequest, and a host of others!"

Regular features will include "My Life and Role-Playing" by prominent FRP hobbyists, product reviews, fantasy and sci-fi artwork, new campaigns, gossip, guest editorials, and more!

Different Worlds, appearing bimonthly, is available for \$9 a year from: The Chaosium, P.O. Box 6302, Albany, CA 94706. □



© 1978 The Chaosium

THE DRAGON IS UPSET!

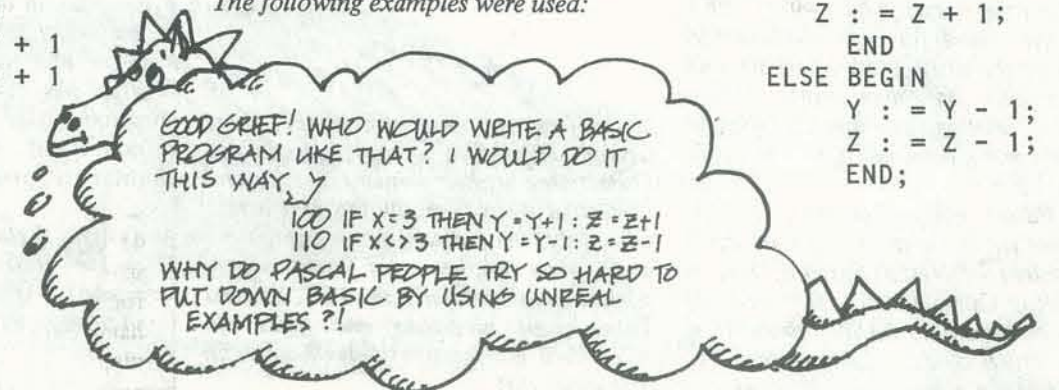
BASIC

```
100 IF X = 3 THEN 500
200 LET Y = Y - 1
300 LET Z = Z - 1
400 GO TO 700
500 LET Y = Y + 1
600 LET Z = Z + 1
700 REM
```

The programming (?) examples here appeared in Northwest Computer News, 3-11, Dec. 1978, page 3, in an article by Joe Felsenstein, titled "Back to BASIC." The article attempted to show how PASCAL is obviously better than BASIC. The following examples were used:

PASCAL

```
IF X = 3
THEN BEGIN
  Y := Y + 1;
  Z := Z + 1;
END
ELSE BEGIN
  Y := Y - 1;
  Z := Z - 1;
END;
```

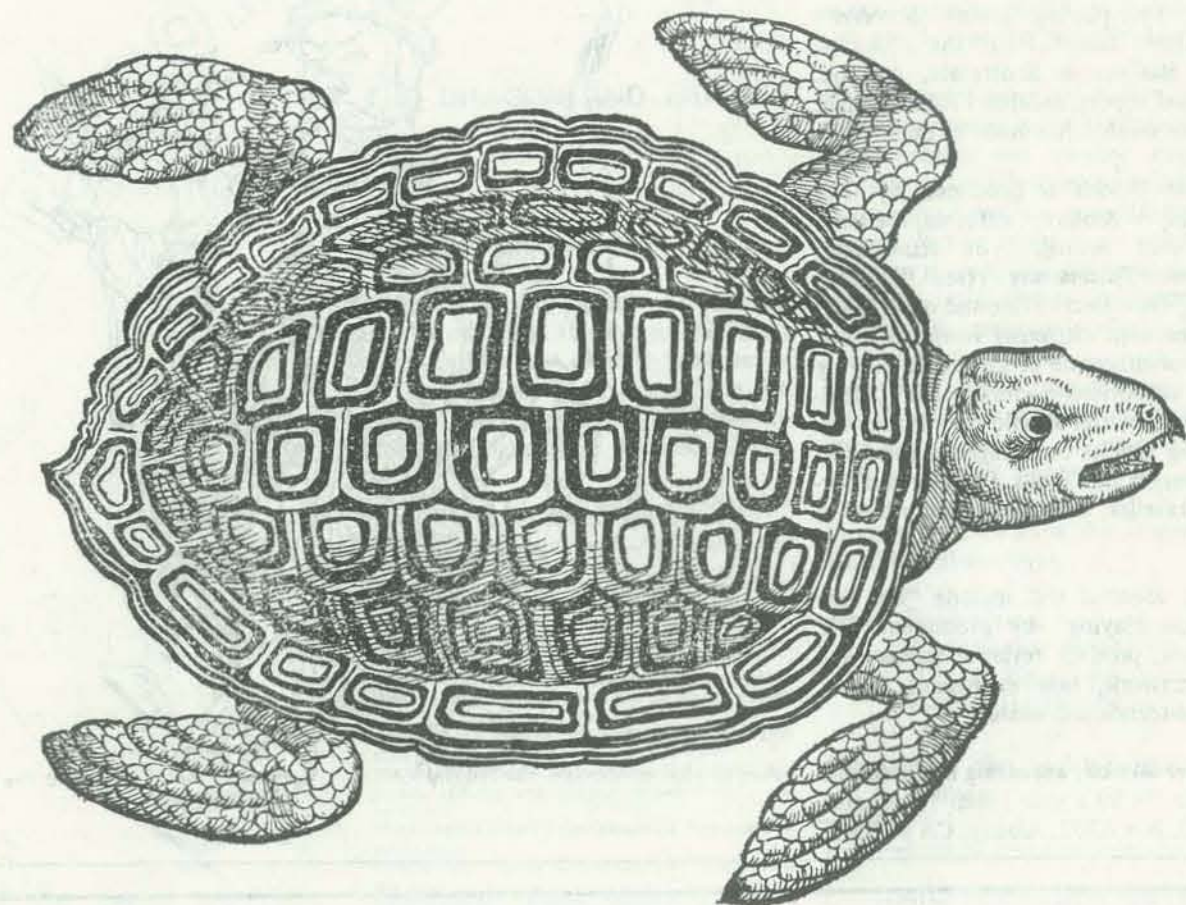


FREE! CIE TRS-80 BULLETIN

This educational bulletin is free from Computer Information Exchange, a non-profit corporation dedicated to free or inexpensive good things for people everywhere. CIE is one of the best sources of information for people who wish to learn how to use, program and enjoy computers. Write for a free copy or even a free subscription. (Do it even if you don't have a TRS-80 or even if you don't yet have a computer!)

From: Computer Information Exchange
 P.O. Box 158
 San Luis Rey, CA 92068

Price: FREE



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

SPOT is devoted to the host of applications—routine and wild—which PET users have found for their machines, as well as to the nitty-gritty of repairs and modifications. In other words, almost anything relating to the PET is fit material for this column.

When Editor Harry Saal isn't tracking down hot items for these pages, he serves as president of Nestar Systems, Inc. in Palo Alto, Calif. Nestar's first product is Cluster/One, a BASIC timesharing system which connects up to 15 microcomputers via a high-speed parallel data

SPOT

The Society of PET Owners and Trainers

EDITED BY HARRY SAAL

bus. Fifteen members of PUG (Pet Users Group, Palo Alto area) plugged into Cluster/One at their January meeting and computed away happily for two hours. Obviously, the creator of this revolutionary system knows a lot about microcomputers, particularly the PET. So send Harry your questions and ideas c/o PCC. He'll give each of them his careful attention—LB

HEARD AROUND THE QUAYSIDE . . .

It looks like the PET family is growing. Commodore was scheduled to announce significant extensions to the PET product line at the Consumer Electronics Show, held in Las Vegas in January. New computers, printers, and the long-awaited floppy disk drive were unveiled.

A new version of the PET is on the way. It has a real keyboard, additional RAM memory, and room for more ROM. The BASIC has been fixed to remove numerous bugs present in the initial version, and the machine language monitor is present in ROM as well.

Actually there are four models: 2001-16N (with a PET graphics-engraved keyboard), the 2001-16B (with a standard typewriter-like keyboard), each with 16K of RAM, and the 2001-32B and N, with 32K of RAM. Both have

numeric pads as well, but do not have the built-in cassette unit present in the 2001-8 model. The external cassette is currently available for \$95 from Commodore. The 16K model will cost \$995, and the 32K version \$1195; delivery was to begin in February.

It is expected that the corrected set of ROMs will be available for use in the PET 2001-8 model, but when and at what price could not be ascertained.

Floppy disk drives are here, too. The model 2040 contains two drives, using "minifloppy" diskettes (five-inch size), each storing about 170K bytes of programs and data. Commodore's non-standard technique for recording employs a variable number of sectors per track, depending on the radius. The disk drive connects to the IEEE bus, and contains two microprocessors of its own—a 6502 (like the PET) and a 6504. It contains its own memory buffer, and uses no additional memory from the basic 8K PET. The dual-drive model costs \$1095, and Commodore plans to have a single-drive model (the 2401) available for \$595, starting in May. The dual-drive 2040 should be available this month (March).

The status of the many-times-delayed 2020 printer has also been resolved. It has been withdrawn, apparently because of problems with the print mechanism supplied by the vendor, but is supplanted by three new printers, offering a variety of options. Each connects to the IEEE port, and contains its own microprocessor, used for both control and intelligent functions, like formatting. The 2021 prints on special electrosensitive paper, at low speed, and costs \$549. It, like the other two models, uses a dot matrix to generate the full PET character set, including graphics.

The 2022 is the top-of-the-line printer, printing on plain paper, with a tractor-feed mechanism. It has a bi-directional printing mechanism, printing 84 lines per minute. It costs \$995. The 2023 is a similar printer, but has no tractor feed, and prints in only one direction, giving about 70 lines per minute. It costs \$849. Orders for the aborted 2020 model can either be withdrawn or switched to one of the new models. The backlog of 2020 orders meant delaying new printer orders; first shipments were scheduled for March.

It should be emphasized that "availability dates" almost always refer to first shipments. The new printers won't be there in quantity, so actual delivery dates will stretch out for most people. And for a popular product, it is usually impossible to have enough units to satisfy the demand (consider the infamous Simon this Christmas). Also, these dates assume Commodore sticking to schedules and having no glitches and delays.

COMMODORE ANNOUNCES PET USER MANUAL

Yes, Virginia, there is a PET User Manual. After a year of delays, the long-awaited, promised user documentation on the PET is ready. The manual is about 250 pages long and contains a pretty good introduction to BASIC and the special features of the PET. The book has nine chapters, with something valuable for all users, from beginners to advanced programmers. It incorporates the information previously issued as separate flyers along with new information. Due to the intermixing of technical and elementary material throughout the book, it is not easy reading, and doesn't have the excellent tutorial style of, say, the TIS Workbooks (read on).

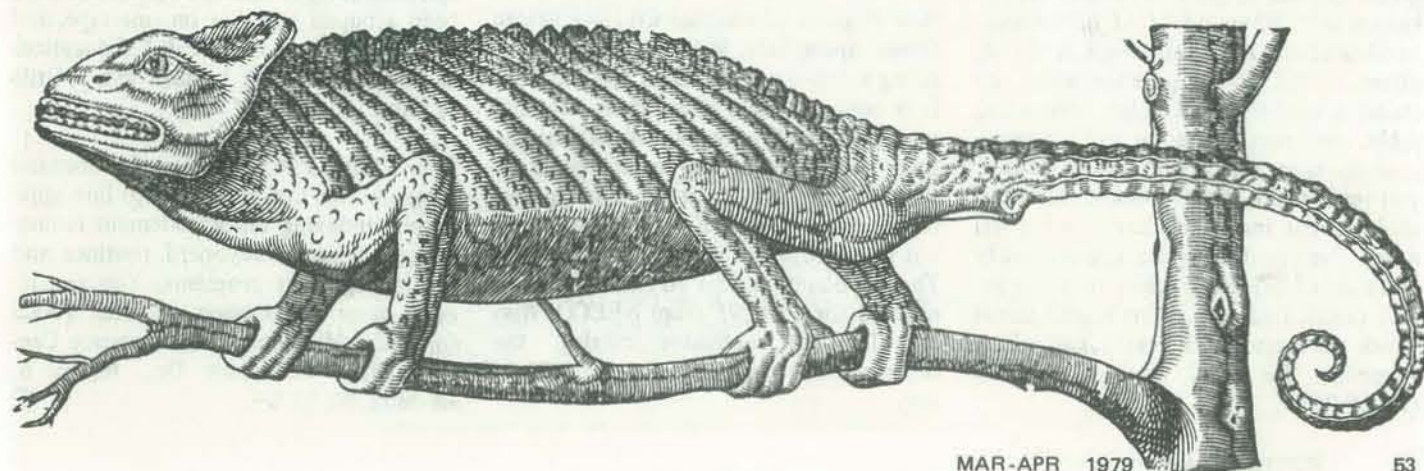
Why don't you have it yet? Commodore is selling the manual (for \$9.95), and including it with new PETs. But if you already have a PET and want a copy of the PET User Manual, write to Commodore Business Machines, Inc., 901 California Ave., Palo Alto, CA 94304, giving Commodore the serial number of your machine (see sticky label on back panel) and the name of the dealer from whom you purchased it. Enclose a check for \$3.50 for shipping and handling. Your local dealer may be able to sell you the manual, but Commodore headquarters is a sure source.

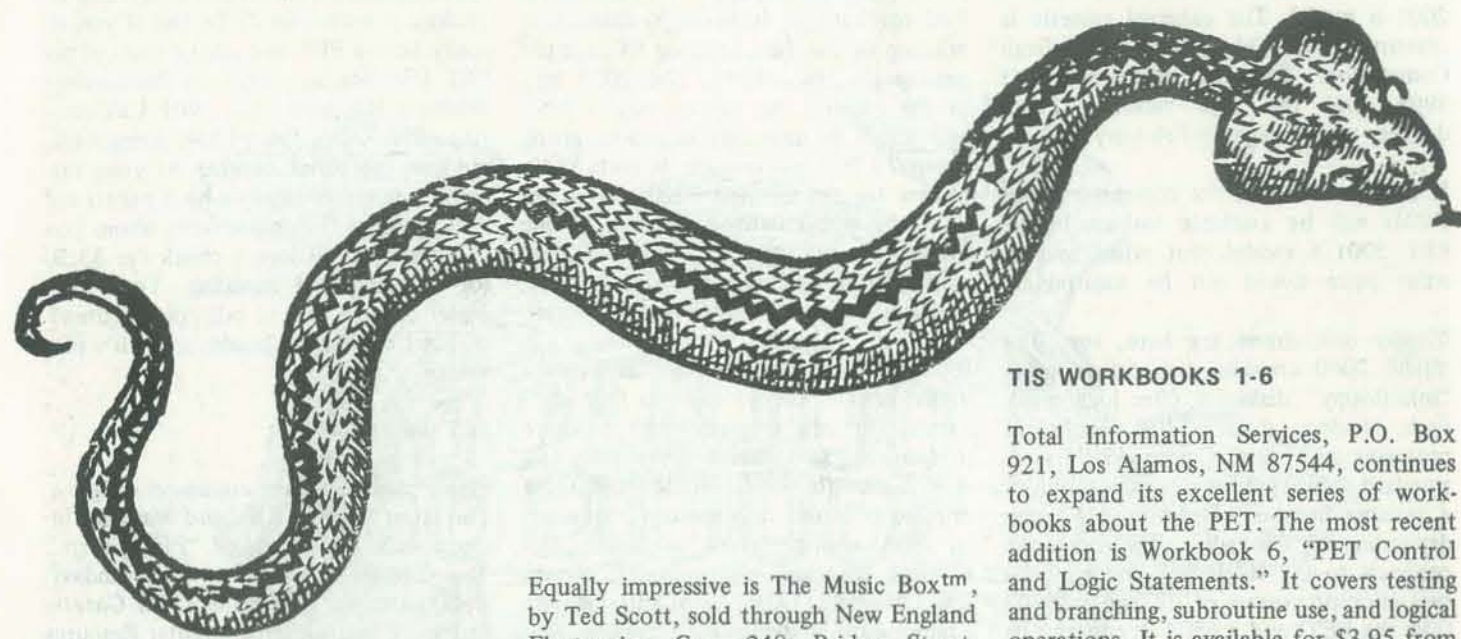
PET IN PRINT

PET's press coverage continues to grow. The latest issue of *Kilobaud Magazine* introduces a feature called "PET-pourri." The column is edited by Len Lindsay, well known for his excellent *PET Gazette* and work at the Microcomputer Resource Center. *Creative Computing* also has begun a regular column dedicated to the PET; it's called "Personal Electronic Transactions." It is authored by Gregory Yob, who not only is quite knowledgeable about the PET, but writes well too!

And the January 1979 issue of *Dr. Dobb's Journal* contains an article by Bill Seiler, author of the Renumber program discussed in our Jan-Feb "SPOT." Bill's article in *DDJ* not only describes how Renumber works, but provides the entire source code for those who wish to follow it in detail, or make modifications.

Those of you who would like to know more about the "guts" of the PET, and how to get access to machine language programs in the Commodore ROMs, such as printing on the screen, saving directly to tape, etc., will find the *PET Assembler Programming Guide* contains a wealth of





handy information which is unlikely ever to be printed by Commodore. This interesting 50-page manual can be obtained from Computhink, 701 Welch Road, Suite 1119, Palo Alto, CA 94304 for \$19.95.

MUSIC FOR YOUR PET

Lately, more and more programs are incorporating sound along with excellent PET graphics. There are several ways this is being done, and in a future issue we'll try to explain them for you. The simplest way to get started right now is to buy one of two excellent sound packages available "ready to go." This avoids the hassle of locating PET-compatible connectors, and gives you a couple of good demonstrations to get started with.

The simplest way to generate sound uses some fancy features of one of the integrated circuits in the PET, and is usually known as "CB2 sound." CAP Electronics, 1884 Shulman Ave., San Jose, CA 95154, offers a PET Sound package which includes a modified AM radio, connecting cable, user port connector and a demonstration tape for \$19.95. From BASIC you can make laser sounds, bounces, clicks, sirens and four octaves of musical notes. I've heard examples, and it is really quite neat (and the graphics they use are well done). CAP also offers several games which use sound (\$4.95 each), and a Tiny Assembler for 6502 machine language (for \$9.95).

Equally impressive is The Music Box™, by Ted Scott, sold through New England Electronics Co., 248 Bridge Street, Springfield, MA 01103. This package includes all the electronics necessary to make your PET play music, and can be placed right inside the PET. (Unfortunately, it uses a quite non-standard scheme for music output which I hope they will convert to the standard CB2 output line.) The Music Box actually lets you compose music on a conventional musical staff drawn on the screen. You can build up to 90 pages of music, with 16 notes per page. You can play any sequence of pages back and also edit them, transpose them, and save them on cassette tape or in memory for incorporation in programs. The visual effects are amazing, as the set-up draws the notes as it plays them, including sharps, flats, rests, etc. The Music Box is available for \$49.95 from NEECO.

FULL-SIZED TYPEWRITER KEYBOARD FOR THE PET

New England Electronics Co., 248 Bridge Street, Springfield, MA 01103, is now offering a full-sized keyboard for the PET. It is contained in a metal enclosure, and permits touch typing, as well as retaining single key function for cursor controls, RUN, STOP, etc. It attaches in parallel to the standard PET Keyboard, and does not interfere with the user or IEEE ports. The keyboard, model NPK-101, can be ordered for \$139.95 from NEECO. Also ask for the extensive catalog, the *Software/Hardware PET Products Directory*.

TIS WORKBOOKS 1-6

Total Information Services, P.O. Box 921, Los Alamos, NM 87544, continues to expand its excellent series of workbooks about the PET. The most recent addition is Workbook 6, "PET Control and Logic Statements." It covers testing and branching, subroutine use, and logical operations. It is available for \$3.95 from TIS. Previous workbooks cover fundamentals of PET BASIC, string and array handling, graphics, cassette I/O, and other miscellaneous features. Workbooks are either \$3.95 or \$4.95, and TIS offers a money-back guarantee. If you are not satisfied with their products, you may return them within 15 days and have your money refunded.

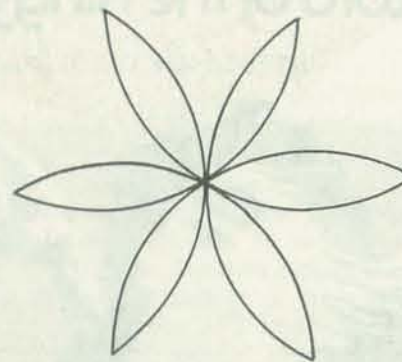
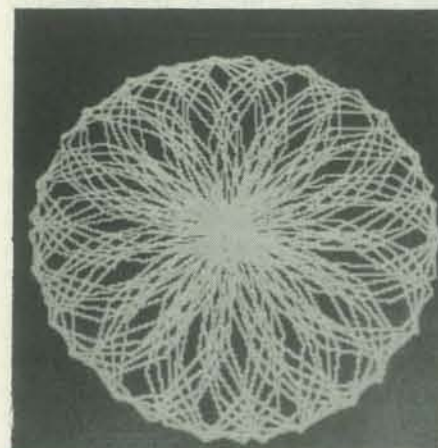
PET GAZETTE PROGRAM PAKS

The *PET Gazette* is now providing an additional service beyond the one-for-one exchange offer for programs donated to the PET Program Exchange. They are offering packages of six programs to anyone (no exchange required) for \$10. (There's a copying charge of \$1 for each program, \$2 for a high quality AGFA tape, \$1 for box and postage, and \$1 handling fee.) A variety of programs have been grouped together on one tape, and are available as: Games Pak 1, Educational Pak 1, Useful Pak 1, Music Pak 1, Utility Pak 1, and Best of Pak 1.

The Utility Pak, for example, contains programs for line erasing, auto line numbering, indexing tapes, statement renumbering, dynamic keyboard routines and file management programs. You can receive more information, or order PAKS from the Microcomputer Resource Center, 1929 Northport Dr., Room 6, Madison, WI 53704. □

Apple·Rose

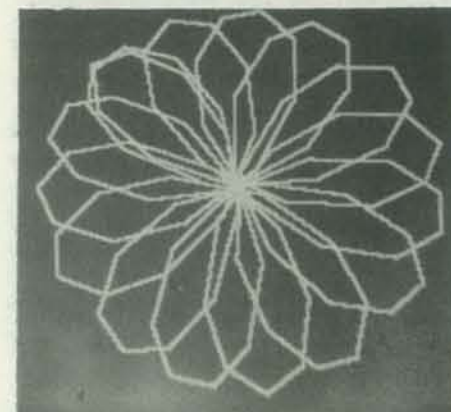
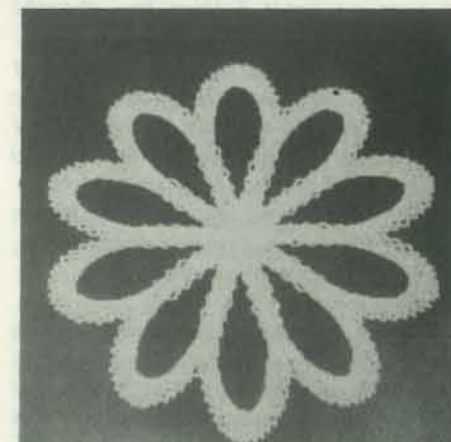
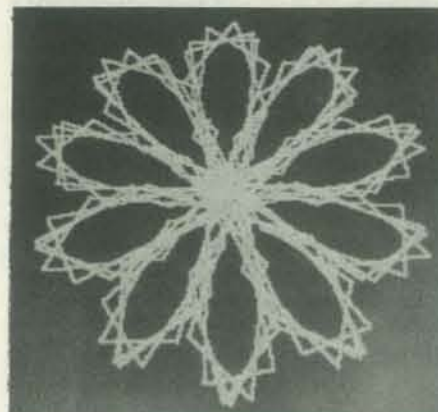
BY JIM DAY



This graphics program, submitted to us by Jim Day of Granada Hills, CA, runs under Applesoft II BASIC. It plots the "rose-leaf" curve shown in the diagram.

To use high-resolution graphics with the Applesoft cassette you need at least 24K. To run in 24K, set HIMEM: 24576 before loading Applesoft. Next load and RUN Applesoft, reset HIMEM: 16384 and then key in and RUN the ROSE program.

The program could be made interactive by changing lines 140 and 150 to INPUT the values of M and N. —CB □



```

100 REM *** APPLE-ROSE ***
110 REM
120 HCOLOR = 3
130 REM SET RANDOM PARAMETERS
140 M = RND(0)*30 + 3
150 N = RND(1)*M + 1
160 REM CENTER DISPLAY
170 HGR2 : HPLLOT 140, 95
180 REM PLOT FOR 256 RADIANS
190 FOR A = 0 TO 256 STEP RND(1) + .01
200 REM FIND RADIUS VECTOR
210 R = SIN((M/N)*A)*95
220 REM CONVERT TO X AND Y
230 HPLLOT TO 140 + R*SIN(A), 95 + R*COS(A)
240 NEXT A : GOTO 140
250 END

```

Reviews

The Movie Lord of the Rings

A year ago most Tolkien fans thought the movie couldn't be done. Today, many would say it hasn't been, at least not with sufficient faithfulness to the original. But as reviewer Michael Madaj points out, Ralph Bakshi's film of Lord of the Rings is a heroic attempt at bringing this complex, incredibly intricate epic to the screen. For this reason alone, it probably deserves a viewing.

The film opened in November and is now showing at theaters nationwide. Hundreds of reviews have appeared, but we felt there was justification for a special forum here. After all, many RC readers are hardcore fantasy freaks. And Tolkien is the 20th century master of the genre.

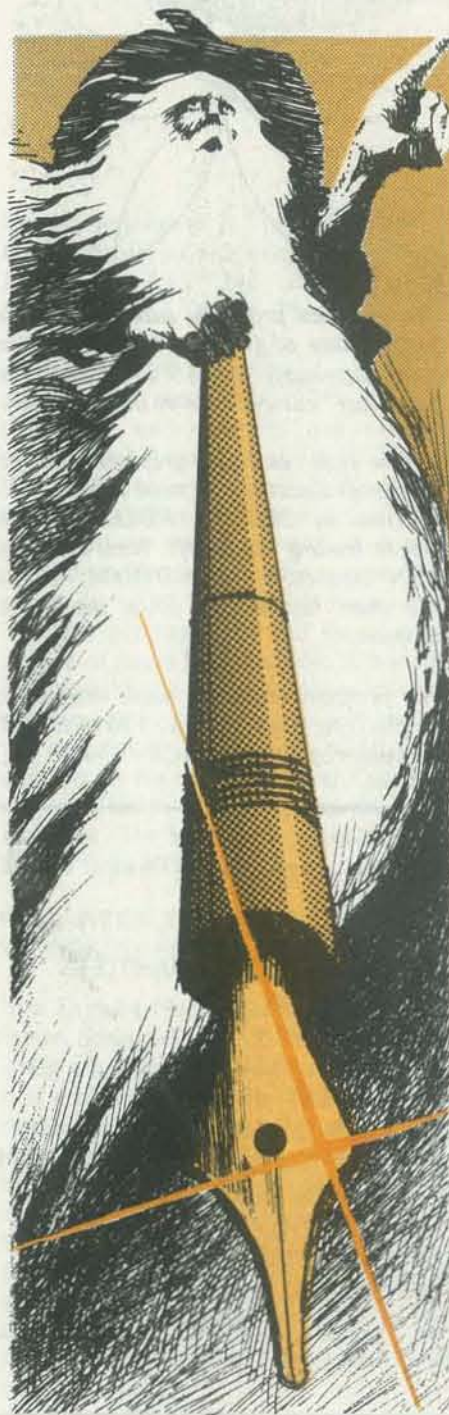
We approached three local Tolkienites for reviews. Each has read the trilogy more than once (reviewer Vershen, six times) and knows the characters and chronology cold, as you will see. None had been to the film at the time we solicited their help, so we didn't know what we'd get. The results were happily diverse.

Michael Madaj is circulation manager for People's Computer Company, Eryk Vershen is a former PCC employee and frequent contributor to RC, and James Moore is a newcomer to these pages. We appreciate their opinions, and welcome your reactions. Agree with them, attack them, offer a counter-interpretation. Just write us a letter. —LB

IT GROWS ON YOU

The more I see it, the more I like it. That's my conclusion after two viewings. Perhaps the second time I relaxed my subliminal expectations so I could just enjoy the movie, maintaining and expanding my own private fantasy while watching the fantasy of Bakshi et al.

Compared to the earlier televised rendition of *The Hobbit*, the film *The Lord of the Rings* is a highly accurate



portrayal of the first half of the epic story. The film strikes a balance of appeal to the spectrum of age groups by characterizing Frodo Baggins as being in his early 20's rather than as the 50-year-old of the book. However, given the peculiar traits of hobbits, this is perhaps within bounds of artistic license. The soundtrack also reflects this balance, while maintaining a character appropriate to the epic tale.

The introduction is superb, to me more dramatic than the widely acclaimed cartooning techniques. The movie begins with shadows playing on the screen in a manner both clear and mysterious while a dramatic voice recounts the history of the Rings up to the time Bilbo brings the Ruling Ring to the Shire.

Departure from Tolkien's story lies primarily in deletions that shorten the material to "suitable" movie length. The major omission is the recounting of most of the journey from Overhill to Bree, leaving out Farmer Maggot, the Old Forest, the meeting with the elves, Old Man Willow, Tom Bombadil, Goldberry, and the Barrow-wights. This type of editing has minimal impact on the major plot development, but seriously affects character development.

A curious effect of editing, combined with a very real effort to maintain the integrity of Tolkien's wording, is that characters often make profound statements that have a tendency to go flat because they are not well supported by the context. When Aragorn says, after Gandalf's plunge into the abyss, "We must do without hope," I would expect to feel something in response, but I didn't. It comes too fast—without time for reflection or filling in gaps with one's own imagination. An anomaly that stands out in my mind, and this is subjective, is the length of time allotted for Frodo's encounter with the Black Riders at the Ford of Rivendell. The tale is embellished so that Frodo's horse falls and later collides with the Black Riders.

The scene is effective in creating an impression, though.

What were some personal highlights? Gandalf leaping through Balin's book found in the Mines of Moria. His fight with the Balrog: "You cannot pass." And later, his recounting of his descent and resurrection: "...I wandered far on roads I will not tell...Naked I was sent back for a brief time, until my task was done." Worthy of mention are the special effects at Orthanc at the occasion of Gandalf's detention there.

What did I find missing in the movie that I found in the book, other than some characters and subplots? A feeling of reliance or trust in the One, without wishful thinking or pseudo positiveness. The feeling of playing a part in a larger plan of evolution, even though the plan is not apparent. The overwhelmingly high value placed on freedom of choice by the free peoples of Middle-Earth.

The moral of the story is that freedom can be preserved in the face of power only by exercising and affirming freedom. The penalty for projection of power is a loss of freedom, which leads ultimately to a dissolution of identity. The realization of the moral awaits Part Two.

What did I find in the movie? High character ideals—Aragorn's disapproval of Boromir's actions, yet his empathy for the man. Wisdom embodied. Repeated refusal of the wise and true to wield the ring and its attendant power—to use the ring against Sauron would be to become Sauron. The earthiness of the dwarves, the etherealness of the elves.

The most endearing character of the movie, of course, is Gollum—which might be surprising to a reader of the book. Precisely because of his personality defects—whining, sneakiness, hatred, and jealousy—he is overdone and becomes a favorite with the audience. The same thing happens with Mime, the dwarf in Wagner's *Ring of the Nibelung*. In Wagner's case, critics consider it a dramatic error on the part of the master. In both cases, I just enjoy.

My own fantasy version of the epic probably entails a billion-dollar budget and

would be created on a format similar to *Star Wars*. The only acceptable alternative is for me to participate actively in a dramatization—and in such a way that I can see everything that goes on. Until the millennium arrives, Bakshi's version is a pleasant alternative.

Reviewed by James Moore
Palo Alto, CA

FLAWED, BUT STILL POETIC

I've heard it said that the *Ring* trilogy is the real history of our planet. The truth in this somewhat outrageous statement is a matter of perception and interpretation. Tolkien intended no such grand design, of course, but even a casual look through his Appendices is impressive: here is the creation of a world, Middle-earth, strikingly similar to our own. Here is a world created out of myth and symbol by a scholar of Middle English, legend, and the fairy tale. The most fantastic scenes are deeply familiar, stirring up, I think, memories of our collective unconscious.

Deep in our unconscious lie archetypal forms that express themselves in dreams, emotions, creative thought, intuition. The expressions of these forms shape our lives in ways we rarely are aware of consciously. The rationalistic framework that our society insists is "real" can only relate to this archetypal input in old, stylized ways: Tolkien is "fantasy" or "escapist" or "pure imagination." When I experience Gandalf's confrontation with the terrible Balrog or Frodo's test at Galadriel's mirror or Boromir's last defense, the thrill up my spine and the tears in my eyes cause me to wonder just where are we escaping to? What old myths is Tolkien evoking and what truth do they symbolize? Is Numenor really Atlantis?

Because of those fascinating possibilities, you might well understand that my reaction to hearing the trilogy was an animated film was less than lukewarm. Oh, God, more cute and cuddly, toylike figures having an adventure! This great mythic saga reduced to a cartoon! Is Walt Disney Studios next contracting to do *Paradise Lost*? Poor Tolkien, I thought. At least he didn't live to see his life's work thus portrayed.

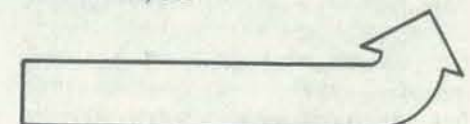
So with considerable cynicism, I went to see the film. What I saw surprised and impressed me. I hadn't known that Ralph Bakshi, director of *Wizards*, was in charge. Or that he had photographed an entire film with live actors and animals and translated it into animation. His surreal landscapes are truly "paintings in motion" and do capture Tolkien's poetic mood.

The movie of *Lord of the Rings* takes us halfway through the trilogy, from Bilbo's birthday party to the battle at Helm's Deep. Final judgment must wait until Part II, but thus far the story is told with accuracy and integrity. For the uninitiated, the *Rings* trilogy tells how the task of casting the One Ring into the fires of Mordor—where dwells Sauron the Dark Lord, who needs only it to be Master of Middle-earth—falls to a band of hobbits, little people who normally prefer "bed and breakfast" to any sort of adventure. As a new age dawns in Middle-earth, the forces of darkness and light are poised and waiting. Darkness has long and secretly been plotting, and the world is almost within its grasp. Enter our little people and our Fellowship of the Ring.

The story tells of a heroic attempt against almost impossible odds. Which is the way I view the movie—a heroic attempt at recreating an ancient tale in a new dimension. But while our Fellowship emerges victorious, I think the film version (thus far) falls short. The hobbits, for all their expressiveness, still are too cute for my taste. Gandalf just doesn't look like the wise and powerful wizard. I miss Tom Bombadil. We tend to skim the surface and linger too long in battle.

Not many who see the movie will ponder the secrets of the ancient myths, but it may eventually lead people back to the books themselves, where the deep and lasting treasure lies. Since there are many of us like Samwise the Hobbit—willing to forsake life and limb just to glimpse an Elf—it is hardly surprising that this film was made. As our own New Age dawns, the unconscious is becoming visible. See you at Findhorn?

Reviewed by Michael Madaj
Menlo Park, CA



**INTERESTING,
BUT NOT RECOMMENDED**

While the animated film *The Lord of the Rings* is, in certain respects, interesting, it does not measure up to Tolkien's work of the same name. There are certainly many superficial similarities between the film and the book, but ultimately the film does not tell the same story. This is not in itself bad. However, the film suffers by comparison, and confirmed Tolkien fans will find much to scream about.

A brief rundown on the changes, omissions and errors in the film for people who know the book:

- About five chapters have been removed from the first book. Most of leaving the Shire, all of Backland, Tom Bombadil and the Barrowdowns are left out. There are also innumerable minor changes and omissions. For example, Gandalf's flash of light when Bilbo disappears at his party is left out; Legolas rather than Glorfindel meets Frodo & Co. on their way to Rivendell; the fight with wolves on the way to Moria is omitted; the whole Legolas-Gemli rivalry is left out; and so on.

- The characterizations are almost all inconsistent with those in the book. Motivations become obscure.

- The main fault of the film—apart from having diminished the story line—is its visual technique. Three techniques are used—full animation, lightly painted-in film and something that looks like color filters with other jazz. The intermingling of methods gives the film a rather uneven feel, compounded by the portrayal of certain characters (e.g. the Black Riders) in different techniques at different times. This does nothing for continuity—and must be very confusing if you don't know the story.

- Plenty of things and people look wrong: Orthanc, Samwise, Galadriel, the Orcs. The last two especially are very wrong; I mean Galadriel doesn't even look pretty!

However, the visual aspects are also what made the film somewhat interesting. The techniques have some potential, even if they did not work well here. The pure animation seemed to go downhill as the film went along, with some aspects of the

final scene (the Battle of Helm's Deep) coming across as crude and amateurish—at least to me. On the whole I would not recommend this movie.

Reviewed by Eryk Vershen
Palo Alto, CA

**THE KID'S REVIEW:
VIDEOBRAIN**

VideoBrain Computer Co.
2950 Patrick Henry Drive
Santa Clara, CA 95050
(408) 988-3020

OLD DAD'S WORDS

Phyllis Cole, in the May-June 1978 issue of *PC*, discussed the VideoBrain computer and some of the cartridge programs that come with the system. We recently received a VideoBrain from Umtech, Inc. (the parent company of VideoBrain Computer Co.) for additional use and review. Since we already had an adult perspective on the product, I decided to let my two sons, ages 8 and 9, use the system for about a month and then solicit their reactions. Their responses to the questions of what they liked/disliked about the computer follow.

It is interesting that all of their reactions deal totally with the software packages of the system. They never mention particulars about the machine itself—the joysticks, the keyboard or the fact that it interfaces with the television set. In fact, all I did was connect the machine to the TV and let them and their friends start using the system. I did not tell them anything about the packages, how the joysticks worked or other such detail. They discovered all of that information by working with the machine.

Each software package has a printed user guide. The children sometimes refer to the guides when they first try a package. They seldom read much of the data in the guide. More often, they simply type a response to the request for an input and see what happens. If they have difficulty learning how the game works by entering responses, they turn to the guide. If the guide is too complex, they go to another game or package.

One last comment before we get to the children's remarks: the order of their responses indicates which programs they

play more often. The first program on the list, *ViceVersa*, is played a lot; the last programs hardly at all.—RZ

THE KIDS SPEAK

We like *ViceVersa*. It is a fun game. It gives you lots of different choices for the games. Hard games. Easy games. You can make the game as hard as you want. The really hard game takes a long time to play because the computer has to think a lot between each move. We like it also because you can play it with another person.

Gladiator is fun. Some of it is. It has three different games. Space Shootout is the best. The ships are easy to hit and to move around. Football is okay. But it is hard to play. It is hard to complete passes. The Archery game is not much fun.

Lemonade Stand is okay to play. Lots of us can play at once. The noises it makes are neat. Especially the circus.

VideoArtist lets you draw stuff all over the screen. It also has one part where the computer draws stuff.

The *Music* program is hard to use. The part where you make up music is hard to use. The easy part is where you copy a song the computer plays. But it only has two songs to copy. It needs more.

Checkers took a while to understand how to play it. We only played a couple of times. We lost the games. You can only play against the computer. Can't play with a friend.

Pinball has lots of games. Sometimes the ball gets stuck. The score gets really big. We have to shut it off to stop it.

We didn't play *Blackjack* much. *Tennis* was a hard game to play.

The *Math* and *Word* games were hard to figure out what to do. They were not much fun.

We want more games where we can play the computer or each other. Also more like *ViceVersa* where you can make it as hard as you want. It is fun. Are there any new games?

Reviewed by Jjago Zamora, age 8, and Fante Zamora, age 9. □

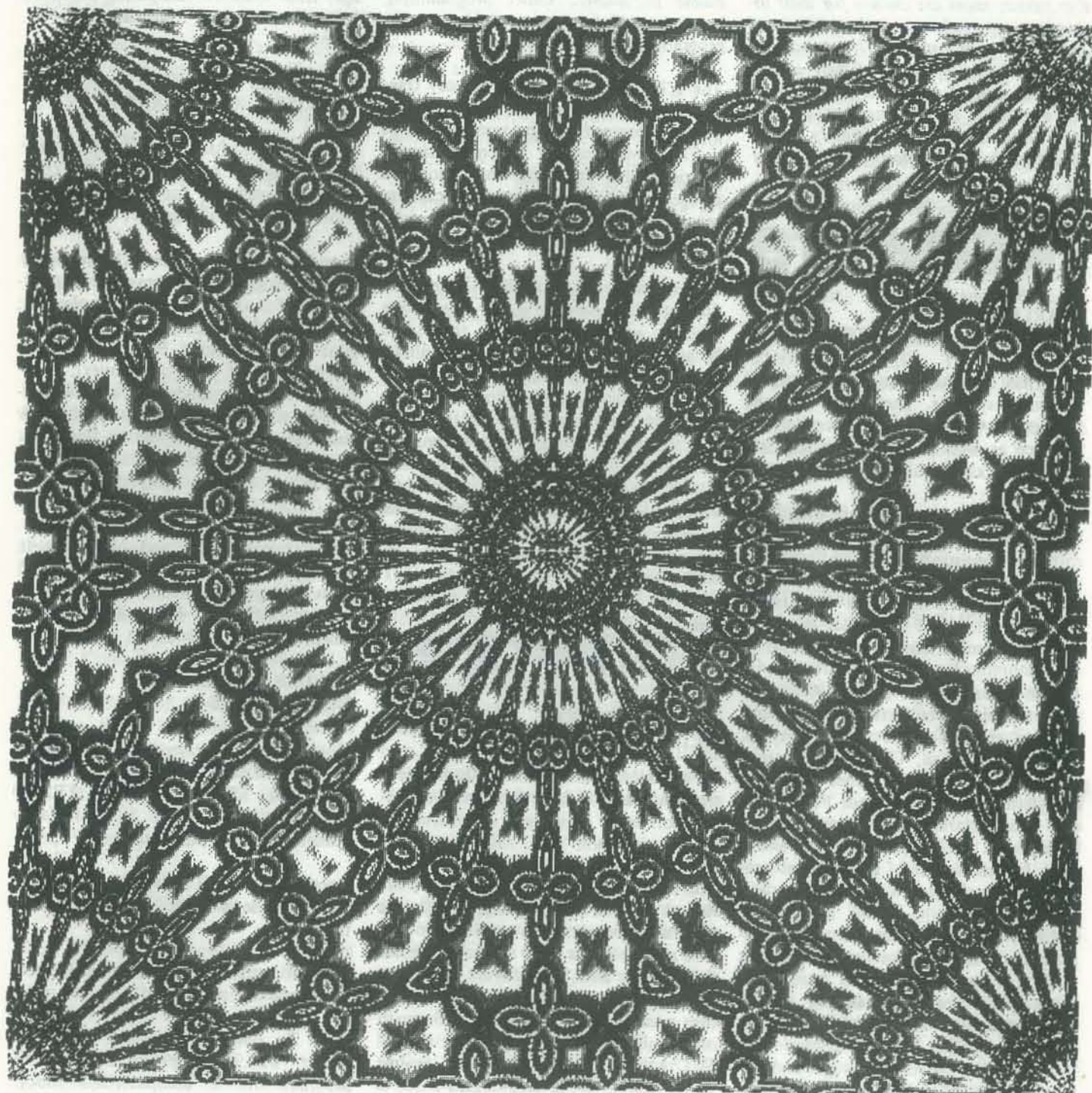
Color 'Your Own' Graphics

BY
DANIEL SILVA

The cover this issue is one of the many intricate computer generated graphics by Dan Silva, a friend of the dragons. Take the cover and this page, make copies of them on a copier and use colored markers to color in the open areas. You will find that you can work quite freely because the dense dark

areas hide mistakes. Try the same design with several different color combinations. Once colored, each finished graphic is unique. Dan is developing a complete coloring book based on these designs. He plans to submit an article describing how the patterns are being generated.—RZ □

©1979 by Daniel Silva



Announcements

Edited By LeRoy Finkel

The items in this section are culled from the many press releases and product announcements we receive each day. While some selections are based on the whim of this editor, most are chosen for their interest to our readers, namely recreational and home applications. The words are those of the news source, slightly edited. The inclusion of an announcement does not constitute an endorsement by People's Computer Company, merely a posting of timely information. —LF

Hardware

ATARI INTRODUCES TWO PERSONAL COMPUTERS

Atari has introduced two new personal computer systems. They were designed for people with no prior computer experience as well as those with sophisticated needs and requirements.

The Atari line of personal computers will have a substantial library of computer software consisting of applications such as personal financial management, income tax preparation, household and office record-keeping and computer-aided instruction in more than 20 subject areas.

In the entertainment sphere, Atari will offer a sophisticated series of action and thinking games for one to four players, such as Basketball, Chess, Life, and a variety of simulation games including Kingdom, Lemonade Stand, Fur Trader and Stock Market. The Atari program library will be continuously expanded by a full-time staff of professional programmers.

The new Atari computers feature custom-integrated circuits for color graphics display, superior sound and music synthesis, slots for instantaneous use of preprogrammed solid-state cartridges, and compatibility with a custom tape recorder for program storage and retrieval. They are UL approved and con-

nect directly to a standard color or black-and-white television.

Both the ATARI-400™ (the general purpose system) and the ATARI-800™ (the specialized system) are programmable in BASIC. Other programming languages will become available on pre-programmed solid state cartridges.

The ATARI-400™ allows an easy transition from video games to full-fledged personal computing. The system features a 57-key monopanel keyboard, single-cartridge slot for solid-state programs of up to 8,000 bytes of memory, cassette recorder capability, and an internal audio speaker.

The specialized ATARI-800™ features dual-cartridge capability, user expandable random access memory up to 48,000 bytes, a series of optional peripheral devices including a high-speed floppy disk for mass data storage and retrieval, and a 40-column printer utilizing standard paper. The versatile and expandable nature of the ATARI-800™ system allows consumers to select components tailored to their specialized needs. Other peripheral devices, including telecommunications capabilities, are currently under development.

Software

EXIDY SORCERER GETS GRAPHICS GAMES

Six graphics games for the Exidy Sorcerer are now available from Creative Computing Software. Some are old favorites; others are brand new. *LEM* is a lunar lander with graphics display and optional auto pilot; *Nuclear Reaction*, a non-violent game of skill in which two players alternately bombard an atom with protons and electrons until it reaches critical mass. In *Pie Lob*, two players take turns lobbing custard cream pies at each other over a sand castle. *Bounce* traces the path of a ball bouncing around the screen. In *Checkers*, the computer plays against you at the novice level. *Dodgem* requires

strategy to get your pieces across the board before your opponent does.

Order Sorcerer Graphics Games, Cat. No. CS-5001. The tape cassette, with instruction booklet, costs \$7.95 (plus 75¢ postage) from Creative Computing Software, P.O. Box 789-M, Morristown, N.J., 07960.

NEW GAMES FOR OHIO SCIENTIFIC SUPERBOARD II

A games tape which takes full advantage of the graphics capabilities of the computer is available for Ohio Scientific Superboard II/Challenger 1P.

It contains four games: *Dodgem*—use strategy to get your pieces off the opposite side of the board (one or two players); *Tank Attack*—seek and destroy enemy guns hidden among houses and trees before they get you (one player); *Free-for-all*—airplane, destroyer, and submarine vie to demolish each other (one or two players); *Hidden Maze*—find your way through an invisible maze with one-way gates (one or two players).

Order Cat. no. CS-6001. Tape cassette, instruction booklet, and box liner in hard plastic box available for \$7.95 (plus 75¢ postage) from Creative Computing Software, P.O. Box 789-M, Morristown, NJ, 07960.

HAYDEN ANNOUNCES CASSETTE LINE

Hayden Book Company has released a line of cassette programs compatible with the PET, TRS-80 Level I, TRS-80 Level II, KIM, Apple II, and Exidy Sorcerer personal computers.

Documentation is provided with each tape or in separate books with the same title.

The following programs are available at your local computer store:

How To Build a Computer-controlled Robot, \$14.95. Features five control programs for a computerized robot:

joystick control, self-direction, impact sensor control, ultrasonic detection, and voice recognition. #00100 (KIM).

Game Playing with Basic, Tape 2, \$9.95. Features 10 programs: *Knight's Tour*, *Guess the Number*, *Prime Numbers 1*, *Prime Numbers 2*, *Chinese Remainder Theorem*, *Perfect Numbers*, *Fibonacci Numbers*, *Amicable Numbers*, *Square Numbers*, *Armstrong Numbers*. #00301 (PET), #00302 (TRS-80 Level I), #00303 (TRS-80 Level II), #00304 (Apple II).

Game Playing with Basic, Tape 3, \$9.95. Features 10 programs: *Slot Machines*, *Blackjack*, *Roll the Dice*, *Tower of Hanoi*, *15 Puzzle*, *Buried Treasure*, *Odd Cell Magic Square*, *4x4 Magic Square*, *Magic Square Starting With Any Number*, *Geometric Magic Square*. #00401 (PET), #00403 (TRS-80 Level II), #00404 (Apple II).

Sargon: A Computer Chess Program, \$19.95. Features the complete program that won the 1978 West Coast Computer Faire Chess Tournament. #00603 (TRS-80 Level II).

The First Book of Kim, Tape 1, \$9.95. Features these 14 recreational programs: *Addition*, *Asteroid*, *Bagels*, *Bandit*, *Blackjack*, *Bitz*, *Black Match*, *Card Dealer*, *Chess Clock*, *Clock*, *Code Test*, *Craps*, *Duel*, *Farmer Brown*. #00700 (KIM).

The First Book of Kim, Tape 2, \$9.95. Features these 14 recreational programs: *Hi Lo*, *Horeserace*, *Key Train*, *KIM Nim*, *KIM-Tac-Toe*, *Lunar Lander*, *Multi-Maze*, *Music Box*, *Ping-Pong*, *Quick*, *Reverse*, *Teaser*, *Timer*, *Wumpus*. #00800 (KIM).

The First Book of Kim, Tape 3, \$9.95. Features these 13 utility programs to make your KIM a more powerful machine: *Branch*, *Browse*, *Directory*, *Hyper-tape*, *Memory Test*, *Mini Dis*, *Movit*, *PLL Set*, *Relocate*, *Sort*, *Super Dup*, *Verify Tape*, *Vu Tape*. #00900 (KIM).

AUSTRALIANS OFFER NEW VERSION OF LOGO

The Elizabeth Computing Centre in Tasmania, Australia, is making its RSTS adaptation of LOGO available to other PDP-RSTS users.

The LOGO language was invented by Seymour Papert and his colleagues at

the Artificial Intelligence Laboratory at the Massachusetts Institute of Technology. This version of LOGO is based in Papert's LOGO (which was designed for unusual peripherals) and supports the MIT Turtle, Tektronix 4006-1 and conventional terminals for turtle geometry.

LOGO forms a highly structured programming language based on a small set of primitives which are used by the programmer as building blocks to form complex procedures.

LOGO has been used successfully with primary school students as well as secondary and college levels.

LOGO is written in MACRO 11 and runs under RSTS Version 6C. For more information, contact Sandra Wills or John Gilbert at the following address: Elizabeth Computer Centre, 256-274 Elizabeth Street, Hobart, Tasmania, Australia 7000.

NEW MAILING SYSTEM DESIGNED FOR TRS-80

MAIL-III is a comprehensive mailing list system for Radio Shack's TRS-80 system. It consists of two programs. The first program lets you enter, display, search, update, delete name and address information. It also initializes the mailing list and displays system information, such as the maximum number of records allowed and number of records used.

The second program produces labels sorted in name, city, state or zip code order. Labels can be printed on the printer or displayed on the screen. A two-digit "select" code is used to identify an input session or to classify the people in your mailing list, such as doctors, commercial accounts, paid members, etc. You can print those labels that belong to a certain code or a specified range of codes, such as all the labels entered today.

You can put 500 names in a diskette—more if you have another drive—or use the diskette only as a data file. For an unlimited number of names, the mailing list can go on more than one diskette. The system is easy-to-use, comes with full documentation and step-by-step tutorial.

Diskette and 16K required. \$35. A simplified cassette version requires 16K and Level II BASIC, and is sold for

\$19. Write to: Micro Architect, 96 Dothan St., Arlington, MA 02174.

TEXT EDITORS FROM TSA SOFTWARE

These easy-to-use, on-screen text editors work on the normal video terminal already in your micro-system. Just pop in the disk and you're ready to key in manuscripts or read them from existing disk files. Simple commands allow you a full range of editing options.

DAISY allows you to add, delete or change the text by moving the cursor to the appropriate location, giving a simple command (often just one character), and typing in the change.

WPDAISY is the word processing version of this system, which includes both space and proportional justification. WPDAISY allows you to call disk files while formatting and has 26 in-memory buffers. Also included is a mail merge program, useful in producing form letters and labels.

The TSA/OS Version is \$125 for DAISY; \$300 for WPDAISY. The CPM Version is \$175 for DAISY; \$350 for WPDAISY.

Available from: TSA SOFTWARE, INC, 39 Williams Drive, Monroe, CT 06468, (203) 261-7963.

Other

COME ONE, COME ALL TO THE COMPUTER FAIRE

The fourth West Coast Computer Faire has issued a call for speakers, participants, and demonstrations. The Faire will be held in San Francisco's Civic Auditorium, May 11-13.

The Faire is primarily concerned with inexpensive computing for home, business, and industry. Some conference sessions will be directed to end users and address specific applications, while others will be of primary interest to business or technical professionals within the microcomputer industry.

Topics at past Faires have included games and tutorials for novices, educational computing and aids for the physically impaired, legal and financial aspects of computer manufacturing and retailing,

and state-of-the-art discussions of hardware, software, systems, and peripherals. Each Faire has also included several user group meetings, as well as industry and retailer meetings.

Those wishing to propose a talk or demonstration or organize a conference session on a particular topic should contact the Faire as soon as possible. Speakers should request an Author's Kit, immediately. Those proposing a talk should submit a paper, or lengthy abstract, in a camera-ready format specified by the Faire, no later than March 1. All such papers that are accepted will be published in the Faire's *Conference Proceedings*.

For Author's Kits or further information, please write Computer Faire, Box 1579, Palo Alto, CA 94302, or call (415) 851-7075.

AEDS SEEKS WHIZ KIDS FOR PROGRAMMING CONTEST

Applications are now available for the Association for Educational Data Systems' 1979 Computer Programming Contest. The contest is designed to recognize those students who develop outstanding projects in the field of computer programming. Students in grades 7 through 12 are eligible.

For an application form and further information, please write: AEDS, 1201 Sixteenth St., N.W., Washington, D.C. 20036.

EDUCATIONAL INSTITUTIONS GET COPYRIGHT BREAK

A new service*, which gives non-profit educational institutions the right (license) to reproduce Digital Equipment Corporation manuals and handbooks, is now available to DEC customers. The service grants the copyright licensee the right to reproduce a specified (by the licensee) number of copies of a manual or handbook for a fee of 25% of the book's cost. This new service significantly reduces manual costs for schools and students.

For more information on this important new service from DEC, contact the Educational Sales Specialist nearest you and ask for a copy of "Educational Institution Reproduction Agreement."

* Valid in the United States only.

KIM USER NOTES

KIM-1 User Notes, the original 6502 newsletter with over 2100 subscribers worldwide (now known as *USER NOTES: 6502*), is expanding its coverage to include the Synertek SYM and Rockwell AIM machines. It will offer twice as much information in a brand new format. KIM will, of course, continue to get the most coverage.

Subscriber rates are \$13.00/6 issues (U.S. & Canada 1st class) and \$19.00/6 issues elsewhere. U.S. funds only.

Purchase orders will be accepted if accompanied with payment. Write *USER NOTES: 6502*, Eric C. Rehnke, Publisher, P. O. Box 33093, North Royalton, OH 44133.

U. MASS OFFERS GRAD PROGRAM IN COMPUTER EDUCATION

The University of Massachusetts has a program of graduate study in Instructional Applications of Computers, leading to master's and doctoral degrees. Study is interdisciplinary, typically involving the fields of education, psychology, mathematics and computer science.

Areas of concentration include: Computer-Assisted Instruction; Computer-Managed Instruction; Programming Languages: APL, BASIC, LOGO; Simulation and Gaming; Mathematics Education; Cognitive Science; Artificial Intelligence.

For further information, contact: Professor Howard A. Peelle, Director, Instructional Applications of Computers, School of Education, University of Massachusetts, Amherst, Massachusetts, 01002, (413) 545-0496.

USERS' GROUPS

TRS-80 users in the Redwood Empire—California's Sonoma, Marin, Lake, and Mendocino Counties— have recently formed a group. "Anyone is welcome, whether they own a TRS-80, small system or large," says the announcement. If you're interested, contact John Revelle, 7136 Belita Avenue, Rohnert Park, CA 94928, (707) 528-1464.

PET users in northern Virginia have organized to "exchange software and

techniques, to discuss problems, and to demonstrate software and hardware." Contact Robert Karpen, 2054 Eakins Court, Reston, VA 22091, (703) 860-9116.

In Annandale, Virginia, TRS-80 users meet the last Wednesday of each month to share ideas, programs, and problems. Contact Rod Wright, 8205 Chivalry Road, Annandale, VA 22003, (703) 560-5854.

SMALL SYSTEMS RATE BIG-TIME BOSTON SHOW

Boston's Hynes Auditorium is the site of the Northeast Computer Show, Sept. 28-30. The Northeast Computer Show will be the largest display of computer equipment ever assembled in Boston.

There will be two separate sections to the show: personal computing and business computing. The personal computing section will feature microcomputers, small computer systems, business opportunities, electronic and video games, career and employment opportunities, educational exhibits, free seminars and lectures.

Exhibitors in this section will display the latest in personal computing hardware and software, computerized music synthesizers, computer amusements, computer-generated art, graphics and animation. Dozens of free lectures and seminars will be given by internationally recognized speakers, along with introductory classes for all categories and levels of enthusiasts.

Computer clubs and organizations, including high school and college clubs, will be encouraged to participate. There will also be special exhibits for children.

For more information about exhibiting in the show, call or write Northeast Expositions, Box 678, Brookline Village, MA 02147, (617) 522-4467.

CORRECTION (TRS-80 GUIDE)

In the Nov-Dec '78 issue we announced a *Guide to TRS-80 Information*, available from Florence Huebner of Oak Park, Michigan. The correct prices on the *Guide* are: \$3.30 each for one; \$2.75 each for two-nine copies; \$1.90 for 10 to 50 copies. □

AHAI™

EXPAND
your horizons
with
calculators
Computers

YOUR RESOURCE FOR LEARNING IDEAS,
REVIEWS, PROGRAMS AND CHALLENGES
FOR YOUR CALCULATOR AND COMPUTER

Please enter my subscription for:

1 year (6 issues) \$10 2 years (12 issues) \$18

Surface mail to Canada and all foreign countries - \$17/yr. Airmail rates available.

check enclosed (payable to Dymax)
 Mastercharge Visa/BankAmericard

Card No. _____
Exp. date _____

Name _____
Address _____
City _____ State _____ Zip _____

Mail to: Calculators/Computers Magazine
P.O. Box 310, J5, Menlo Park, CA 94025



QUALITY IS BASIC™



**TRS-80
PET
APPLE**

GIVE YOURSELF QUALITY
FULLY TESTED PROGRAMS

THAT'S RIGHT! FULLY TESTED PROGRAMS ARE PLAIN TO USE. OUR EXPERT STAFF GET THE "BUGS" OUT OF EACH PROGRAM BEFORE YOU PUT IT INTO YOUR COMPUTER. AND WHEN A PROGRAM HAS PASSED OUR FINAL TEST, IT IS PRINTED AUTOMATICALLY, JUST AS IT WAS RUN MINUTES BEFORE IN OUR POWERFUL SYSTEM.

IN EASY-TO-UNDERSTAND BASIC

OF COURSE EVEN A TESTED PROGRAM IS OF NO USE TO YOU IF YOU CAN'T RUN IT IN YOUR SYSTEM. THAT'S WHY WE CHOOSE BASIC AS OUR "MEDIUM OF EXCHANGE." MORE PERSONAL COMPUTERS USE BASIC THAN ANY OTHER LANGUAGE, AND NO WONDER. BASIC IS EASY TO UNDERSTAND! YOU SHOULDN'T HAVE TO STRUGGLE WITH MYSTERIOUS AND TROUBLE-SOME CODES. INSTEAD, BASIC TELLS YOU WHAT EACH STATEMENT MEANS, AND, WHERE THERE MIGHT BE A LITTLE CONFUSION, OUR STAFF ADD EXPLANATIONS IN PLAIN ENGLISH!

ACCLAIMED BY REVIEWERS

"...WELL-BALANCED...GAMES IN THIS QUARTER ISSUE ARE FASCINATING...QUITE EASY TO UNDERSTAND..." - ARNOLD ZIMMERA, EDITOR, PEOPLE'S COMPUTER COMPANY

"...FASCINATING...SMALL JOURNAL FOR SOBERMATED USERS OF COMPUTERS AND PROGRAMMABLE CALCULATORS..." - HENRY HORN, EDITOR, HEWLETT-PACKARD KEYNOTES

"...PROGRAMS IN CALCULATOR LANGUAGE ARE IN BASIC WITH CLEAR EXPLANATIONS! IF YOU ARE INTO CALCULATOR/COMPUTER GAMES, DON'T MISS THIS ONE!" - LORR H. ALLEN, EDITOR AND LORR HARRY R. KRUSE, EDITORS, PPC NEWSGAM

ORDER BY MAIL

SIMPLY MAIL YOUR PERSONAL CHECK OR MONEY ORDER TO THE RECREATIONAL PROGRAMMER, P.O. BOX 2571, KALAMAZOO, MICHIGAN, 49003.

CHARGE BY TELEPHONE

NO PROBLEM, NO EXTRA COST! CALL (616) 743-1946 BETWEEN 10 AND 5 EASTERN STANDARD TIME AND BILL YOUR SUBSCRIPTION TO YOUR VISA OR MASTERCARD CARD.

ANNUAL SUBSCRIPTION RATES

\$12 WITHIN THE US AND APO'S, PPO'S
\$15 FOREIGN
\$24 FOREIGN AIRMAIL

**THE
RECREATIONAL
PROGRAMMER**™

QUALITY BIMONTHLY COMPUTER SOFTWARE™
P.O. Box 2571 Kalamazoo Michigan 49003