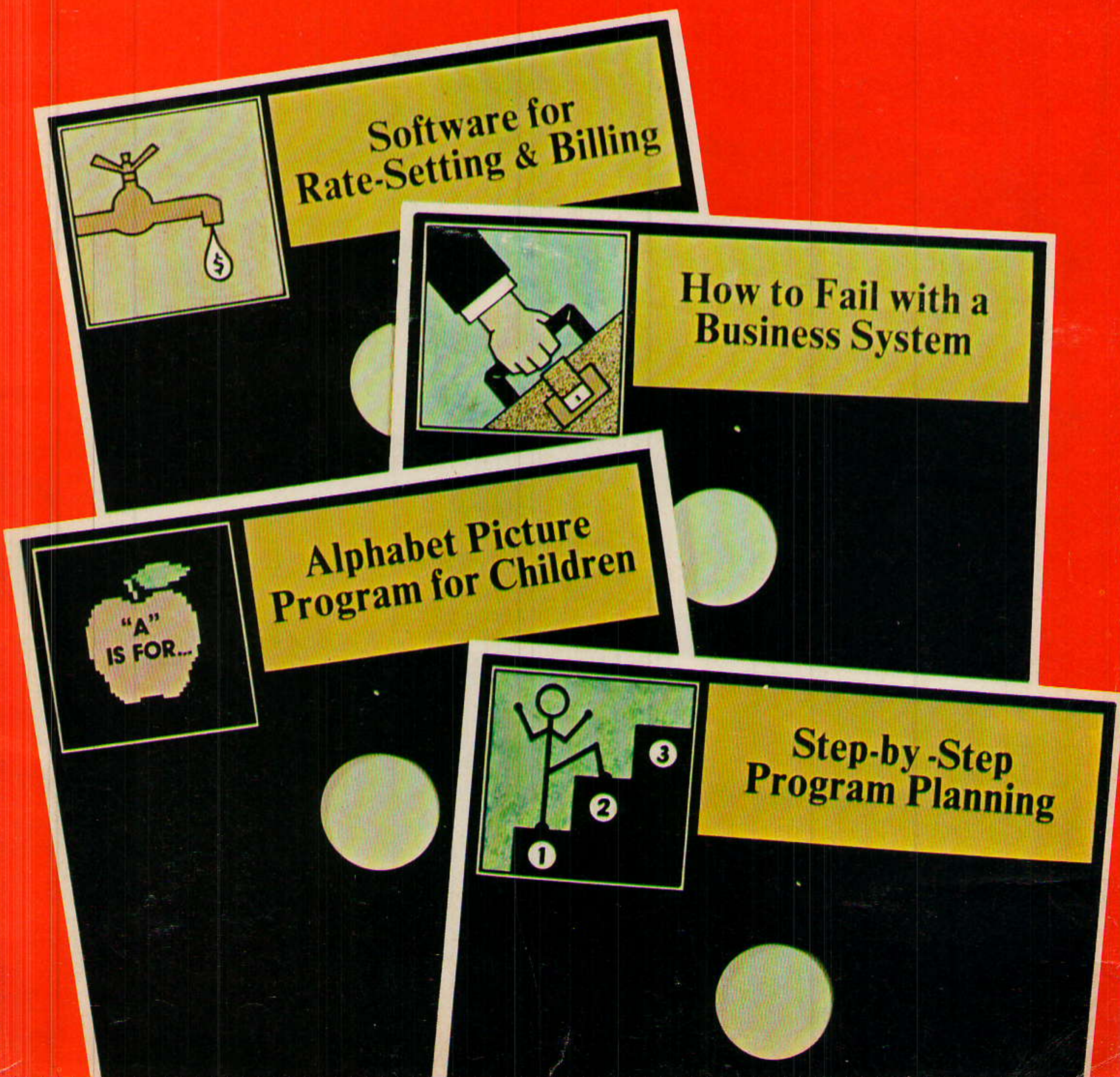
TRS-80 T-BUG • Game-Playing Algorithm • Robots Game

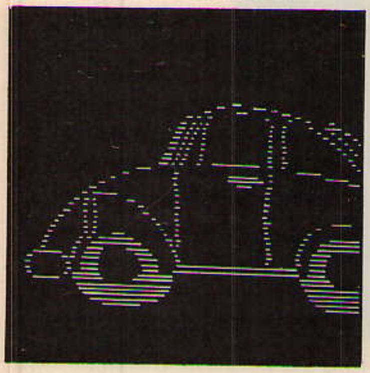
Personal Computing

$2.00
JUNE 1979

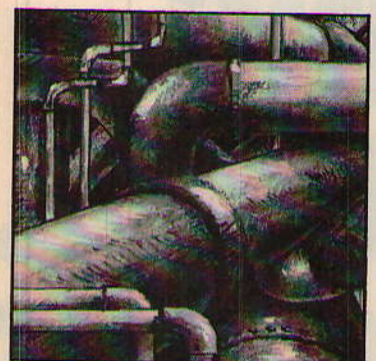


Personal Computing

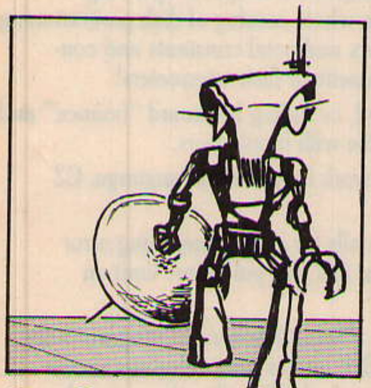
Publication Number USPS 370-770 JUNE 1979 Vol. III No. 6



Page 38



Page 43



Page 60

Cover Design by Stephen C. Fischer

LAUNCHING PAD

DIGGING IN

IN THE MONEY

ON THE LIGHTER SIDE

FUTURE COMPUTING

DEPARTMENTS

© Copyright 1979, Benwill Publishing Corp., a Morgan-Grampian Co.

PASCAL SOFTWARE

COMPLETE BUSINESS ACCOUNTING PACKAGES

ACCOUNTS PAYABLE
ACCOUNTS RECEIVABLE
GENERAL LEDGER
INVENTORY CONTROL
ORDER ENTRY

PS inc

619 N.P. Ave.
Box 2017
Fargo, ND, 58102
(701) 235-8145
DEALER INQUIRIES INVITED
SEE US AT THE NCC

See our other ad in this Magazine

CIRCLE 85

A General Game Playing Program

BY HERBERT L. DERSHEM

Programmers have used the look-ahead strategy to develop competitive game playing programs for games like checkers and chess. A general form of this look-ahead algorithm can be described in terms of a recursive procedure implemented in BASIC for specific games. If your BASIC processor accepts recursive subroutine calls, then you can use this algorithm to play any suitable game by programming three additional subroutines that describe the game. (For more information on recursive programming, see "Recursive Programming in BASIC", April *PC*.)

Consider a game with two players called "computer" and "opponent". At any given point in the game, two descriptors describe the situation: the game status (GS), often the status of the game board; and the player to move next (PM), either "computer" or "opponent". Each GS, PM pair results either in a completed game with a winner, or in a draw, or in a set of legal moves for PM. Each legal move maps the GS,PM pair. Let's consider the case where the players alternate moves, making the new PM generated by a move always different from the previous PM.

Now we're ready to recursively state the look-ahead algorithm which, given a GS,PM pair, evaluates all the legal moves available to player PM and determines the optimal one.

Algorithm Evaluate to find the best move BM for player PM from game status GS with evaluation of E.
Evaluate (GS,PM,E,BM)

1. If(GS,PM) is directly evaluatable, evaluate it and place result in E; return.
2. Generate $MV_1, MV_2, \ldots, MV_n$, the set of all legal moves from (GS,PM), and $GS_1, GS_2, \ldots, GS_n$, the corresponding set of game statuses after the legal moves are applied to GS.
3. If PM = computer, call Evaluate (GS_i, opponent, E_i, BM_i) for i = 1,2,..., n; for E_k, the largest of $E_1, E_2, \ldots, E_n$, set E = E_k, BM = MV_k; return.
4. If PM = opponent, call Evaluate (GS_i, computer, E_i, BM_i) for i = 1, 2,..., n; for E_k, the smallest of $E_1, E_2 \ldots, E_n$, set E = E_k, BM = MV_k; return.

Evaluation of a game status is always from the computer's point of view. The larger the evaluation, the better the status is for the computer. Therefore, the principle behind this algorithm is that the computer always chooses from the legal moves that move resulting in a game status with largest evaluation. On the other hand, the opponent always chooses the move with the smallest evaluation, since that move is the least desirable for the computer.

How does the computer determine whether a move is directly evaluatable? If a game status is terminal, there are no further moves. Or sometimes the computer stops when a certain number of levels of moves have been examined. For example, a 3-level look-ahead will examine all of the computer's legal responses. As you can see, the number of moves that must be examined grows rapidly as the level of the search infinal level (level 3 in the example above), you must implement some heuristic procedure to evaluate the GS, PM pair. The ability of this procedure, the static evaluation function, to accurately evaluate the game's status greatly affects how well the computer will compete. There's a trade-off between the depth of look-ahead and the validity of the static evaluation function. If the static evaluation function is perfect, the computer can use it to evaluate all its alternatives directly and not look ahead at all. On the other hand, if the computer can look ahead clear to the end of the game, examining all of the alternatives, it has no need for a static evaluation function since the perfect evaluation function is the game result: win, lose or draw. In practice we find ourselves somewhere between those two extremes.

For the general BASIC version for this algorithm, see Listing 1. Two additions to the algorithm have been made to speed up the search. Both halt the process when it's obvious no more searching is needed.

Suppose the search is at a level generating the computer's responses. If, at the preceding level, the opponent's best move evaluates to 4 and so far the computer's best move at this level evaluates to 5, why continue the search at this level? The opponent will never choose the current move under consideration because it will evaluate to no smaller than 5 which is already 1 worse than the best move the opponent has examined so far. This condition is tested in line 2100 of the program in Listing 1. In tree searching this process, called alpha-beta pruning, usually saves search time.

Additional savings can result from statement 2130 where, as soon as a player has found a sure winner for himself, he stops searching.

Now let's look at two implementations of the algorithm in Listing 1. The first, found in Listing 2, is the familiar game of tic-tac-toe. The implementation requires the addition of three subroutines to the general game status evaluator at 2000. These are 1000, a move generator; 3000, a static move evaluation function; and 4000, a game-over tester. But the choices shown here are examples: try designing your own improved versions of these subroutines.

The particular implementation here uses a maximum search depth of 10 levels. For tic-tac-toe, this level implies all searches will be terminated by the end of the game since the longest possible game is 9 moves. The static evaluation function returns 100 if the position is a win for the computer,-100 if it's a win for the opponent, and 0 if it's a draw.

Subroutine 2000 has been modified slightly from that shown in Listing 1 to accommodate the presence of only one subscripted variable in Radio Shack Level 1 BASIC, the system on which this program was implemented.

The ancient game of Kalah, our second game, is played on a board with six small pits on either side and large pits at each end. The game begins with 3 markers in each of the small pits as shown in Figure 1.

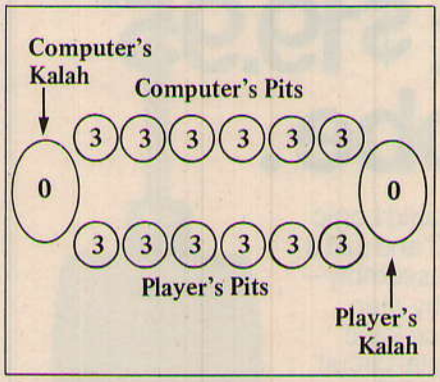


Figure 1 **Initial position of Kalah board**

The players alternate moves according to the following rules:

1. A player moves by choosing a pit on his side of the board and distributing the markers contained in that pit into other pits counterclockwise around the board beginning with the counterclockwise neighbor of the emptied pit. He places one marker in each pit and Kalah in turn until all markers removed are distributed. *Example:* If the opponent began play from the initial board shown above by emptying the fifth pit from the left on his side, after his move the board would look like Figure 2.

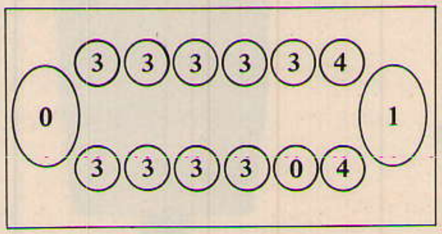


Figure 2 **Example of a move**

P. S. inc.

FOR COMPUTER SOLUTIONS

AR AP GL OE INV

Pascal Software

COMPLETE BUSINESS ACCOUNTING PACKAGES IN

Pascal

You've read about this efficient new block structured language. Now for the first time you can run your business with completely integrated accounting packages written in PASCAL, realizing the speed and efficient file handling capabilities inherent to this language.

GENERAL LEDGER

Allows over a thousand general ledger account numbers. Features a transaction register that forms the AUDIT TRAIL for all transactions. Easy entry and editing of transactions and special routines to prohibit posting of unbalanced transactions that might otherwise go unnoticed.

ACCOUNTS PAYABLE

Allows over a thousand vendors which you can ADD, DELETE or CHANGE. Allows easy entry and editing and the voucher register forms a clear AUDIT TRAIL for your permanent records.

ACCOUNTS RECEIVABLE

Allows for over ten thousand customers (limited by disk storage) easy entry and editing of sales including invoices, finance charges, and credit and debit memos. Also a sales journal that lists all transactions. Cash receipts are easily entered and listed prior to posting. All transactions are posted not only to the ACCOUNTS PAYABLE file but also to the GENERAL LEDGER.

INVENTORY CONTROL
(ORDER ENTRY)

Allows up to 32,000 items, (depending on storage media). Prints invoices, picking tickets and stock status reports. Posts transactions to A/R or A P. Performs sales analysis by product category Automatic updating of inventory file gives you up to the minute stock status. Re-order flags tell you when it is time to order a low stock or out of stock item.

EASY TO USE

All these application programs are menu oriented for ease of operation and minumum personnel orientation and training time. All screens are formatted for clear concise data entry and editing. All totally interactive and easy to comprehend.

AVAILABLE TODAY!

Currently available on floppy disk for the Pascal Microengine —single or double density. Single density Alpha Pascal disks for Alpha-Micro Systems and RX diskettes for PDPll Systems. Also available in single density diskette for any system that runs UCSD Pascal. Also available on 5440 cartridge disks and other media. Write today for our Complete Information Package.

PS inc

619 N.P. Ave.
Box 2017
Fargo, ND, 58102

(701) 235-8145
DEALER INQUIRIES INVITED
SEE US AT THE NCC

CIRCLE 86

2. If the last marker distributed by a player lands in that player's Kalah, the player must empty another pit on his side. This move is called a continuation. The continuation might have another continuation, and so on. *Example:* If the computer's response to the above move was to empty the third pot from the left, it would receive a continuation. This move and its continuation are illustrated in Figure 3.

3. If the last marker distributed on a player's move lands in an empty pit on the player's side of the board, and if some markers are in his opponents pit directly opposite this pit, then the last marker distributed and all the markers in the opposite pit are placed in the Kalah of the player making the move. This move is called a capture. *Example:*If the opponent now empties the lands in the empty pit and captures the computer's four markers on the opposite side (Figure 4).

The winner has the most markers in his Kalah at the end of the game. When a player has no more markers in his pits and it's his turn to move, the game ends. At that point the opponent places all the markers in his pits into his Kalah, and the winner is determined.

Listing 3 shows the application of the game playing algorithm to this game. The continuation complicates matters by requiring two locations to store a move as well as a special coding scheme for continuation moves.

CIRCLE 87

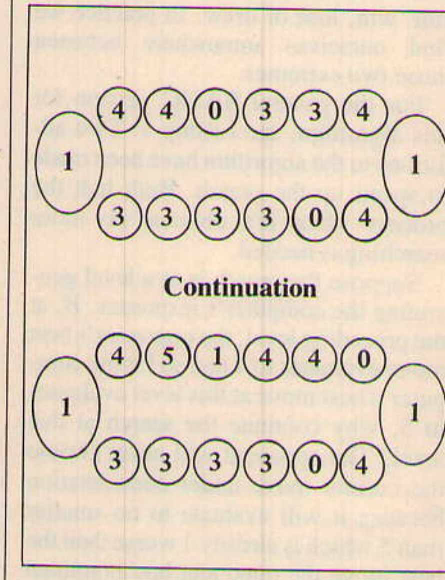


Figure 3 Example of a move and continuation

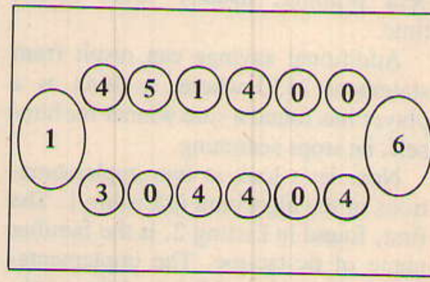


Figure 4 Example of a capture

Now that you've seen these examples, you can implement this algorithm for other games. You might want to improve the computer's performance on these games by providing better static evaluation functions or increasing the maximum depth of search. You must proceed with caution, however. Look-ahead algorithms can consume lots of computer time. So be prepared to wait for the computer's moves. □

Listing 1 – Game-Playing Algorithm

```
1500 REM GENERAL GAME PLAYING PROGRAM.  THIS SUBROUTINE,
1510 REM WHICH IS CALLED BY 1990, WILL ACCEPT A GAME STATUS STORED
1520 REM IN A(1) THRU A(S) AND RETURN IN S(1) THE BEST MOVE FOUND
1530 REM AND IN E THE EVALUATION OF THAT MOVE.
1540 REM PARAMETERS:
1550 REM     S IS THE NUMBER OF LOCATIONS NEEDED TO STORE GAME STATUS.
1560 REM     M IS THE MAXIMUM DEPTH OF SEARCH
1570 REM     N IS A VALUE WHICH IS IMPOSSIBLE CODE FOR A MOVE AND
1575 REM       REPRESENTS A NULL MOVE.
1580 REM     W IS A VALUE SUCH THAT ANY GAME STATUS WHICH EVALUATES
1590 REM       >=W IS A WIN FOR THE COMPUTER AND ANY WHICH EVALUATES
1600 REM       <=-W IS A WIN FOR THE OPPONENT.
1610 REM
1620 REM VARIABLES:
1630 REM     L IS THE LEVEL INDICATOR FOR THE CURRENT LEVEL OF SEARCH.
1640 REM     Z INDICATES PLAYER WHO IS MOVING: 1=COMPUTER,-1=OPPONENT.
1650 REM     Q IS THE STACK POINTER. IT INDICATES THE POSITION IN THE
1660 REM       STACK DIMENSIONED VARIABLE WHERE THE CURRENT GAME STATUS
1670 REM       DESCRIPTION BEGINS.
1680 REM     M(L) IS THE CURRENT MOVE BEING EXAMINED AT LEVEL L
1690 REM     S(L) IS THE BEST MOVE EVALUATED SO FAR AT LEVEL L.
1700 REM     B(L) IS THE EVALUATION OF THE BEST MOVE SO FAR AT LEVEL L.
1710 REM     E IS THE VARIABLE IN WHICH THE EVALUATION OF THE BEST MOVE
1720 REM       IS RETURNED.
1730 REM
1740 REM SUBROUTINES:
1750 REM 1000 GENERATES FROM MOVE M(L), THE NEXT MOVE IN A SEQUENCE
1760 REM   OF ALLOWABLE MOVES FROM THE GAME STATUS STORED AT POSITION
1770 REM   Q IN THE STACK. THE MOVE IS STORED IN M(L), AND THE NEW
1780 REM   GAME STATUS IS PLACED IN THE STACK BEGINNING AT POSITION
1790 REM   Q+S. THE FIRST MOVE IN THE SEQUENCE IS GENERATED WHEN
1800 REM   M(L)=N, THE NULL MOVE, WHEN THE SUBROUTINE IS CALLED.
1810 REM   IF M(L) IS THE LAST MOVE IN THE SEQUENCE, THEN M(L)=N
1815 REM   IS RETURNED.
1820 REM 3000 EVALUATES THE GAME STATUS STORED BEGINNING AT POSITION
1830 REM   Q OF THE STACK USING A STATIC EVALUATION FUNCTION. THE
1840 REM   VALUE IS STORED IN E.
1850 REM 4000 TESTS THE GAME STATUS STORED BEGINNING AT POSITION Q
1860 REM   OF THE STACK.  IF IT IS A GAME ENDING POSITION, THAT IS,
1870 REM   IF NO MORE MOVES ARE POSSIBLE, O IS RETURNED AS 1.
1880 REM   OTHERWISE, O IS RETURNED AS ZERO.
1890 REM
1900 REM THIS SUBROUTINE IS WRITTEN IN RADIO SHACK LEVEL I BASIC
1910 REM EXCEPT FOR THE USE OF EXTRA
     DIMENSIONED VARIABLES M,S, AND L.
1920 REM THESE HAVE BEEN USED FOR CLARITY.
1930 REM
1989 REM INITIALIZE L AND Z ON THE FIRST CALL.
1990 L=0: Z=-1
1999 REM UPDATE L,Q, AND Z FOR THE NEXT LEVEL OF SEARCH.
2000 L=L+1: Q=S*(L-1)+1: Z=-Z
2009 REM TEST IF GAME IS OVER.
2010 GOSUB 4000
2018 REM IF LEVEL IS TO THE MAXIMUM OR GAME IS OVER, EVALUATE
2019 REM USING STATIC EVALUATION FUNCTION AND RETURN.
2020 IF (L<=M)*(O=0) GOTO 2050
2030   GOSUB 3000
2040   GOTO 2150
2049 REM INITIALIZE FOR BEST POSSIBLE MOVE SEARCH.
2050 M(L)=N: S(L)=N: B(L)=-Z*W
2059 REM GENERATE NEXT MOVE
2060 GOSUB 1000
2069 REM IF NO MORE MOVES, SET E AND RETURN.
2070 IF M(L)=N THEN E=B(L): GOTO 2150
2079 REM EVALUATE THIS MOVE, M(L), BY A RECURSIVE CALL.
2080 GOSUB 2000
2086 REM IF THE BEST MOVE AT THIS LEVEL IS ALREADY BETTER FOR Z
2087 REM THAN THE BEST MOVE FROM PRECEDING LEVEL WAS FOR -Z, THEN
2088 REM THIS MOVE WILL NOT BE CHOSEN BY -Z ANYWAY, SO RETURN
2089 REM WITHOUT EVALUATING THE OTHER MOVES AT THIS LEVEL.
2090 IF L=1 GOTO 2110
2100   IF Z*E>=Z*B(L-1) THEN B(L)=E: GOTO 2150
2108 REM IF THIS IS THE FIRST RESPONSE TRIED OR IT IS BETTER
2109 REM THAN THE BEST SO FAR, RECORD IT AS BEST SO FAR.
2110 IF (S(L)<>N)*(Z*E<=Z*B(L)) GOTO 2060
2120   B(L)=E: S(L)=M(L)
2129 REM IF THIS RESPONSE WINS, THERE IS NO NEED TO SEARCH MORE.
2130 IF Z*B(L)<W GOTO 2060
2149 REM ADJUST L,Q, AND Z AND RETURN.
2150 L=L-1: Q=Q-S: Z=-Z: RETURN
```


HOBBY WORLD® CALL TOLL FREE: (800) 423-5387
CA, HI, AK: (213) 886-9200

TRS-80 SOFTWARE CASSETTES

Cat No.	Description	Price
1093	SARGON CHESS, II.	19.95
1041	STAR TREK, II.	14.95
1036	SCI FI GAME SAMPLER, I/II.	5.95
1042	TAROT I/II	5.95
1179	CRIBBAGE I/II.	9.95
1192	REAL TIME LUNAR LANDER II.	7.95
1195	BRIDGE CHALLENGER II.	14.95
1186	AIR RAID I/II.	14.95
1187	PILOT I/II	14.95
1047	OTHELLO I/II	5.95
1043	SMALL BUSINESS BOOKKEEPING I/II.	14.95
1051	DAILY BIORHYTHM PROGRAM I/II	5.95
1049	MICRO TEXT EDITOR I/II	9.95
1038	INVENTORY MODULAR I/II.	19.95
1153	EDIT-80, text editor II (32K)	39.95

ANADEX PRINTER Model DP-8000 $999

Connects easily to most popular computers including TRS-80. 3 basis ASCII compatible interface configurations are provided. 80 columns, 112 cps. 84 lines per min, bi directional printing. Out-of-paper detector, uses standard low-cost papers. 96 character set, 9x7 dot matrix characters. Original plus up to 3 copies. Skip-over-perforation control, double width printing, 8 programmable vertical tab positions. Excellent readability. Superior to other printers costing three times as much.

TRS-80 ELECTRIC PENCIL

Character oriented word processing system. Produce mailing lists, business forms, large numbers of original correspondance, camera ready copy for printing...all on your TRS-80. No carriage returns or hyphenations, line formatting is done by the Electric Pencil! Also features right margin justifying, page numbering, and tilting, and many combinations of line length, page length, etc. For TRS-80 level 1 or 2, 16K, and virtually any printer.

$95

TRS-80 LEVEL III BASIC $42

As advertised in March Interface. Loads on top of level II, turns your TRS-80 into a powerful system. Solves loading problems, cures keyboard "bounce". Software cassette, has the power of a hardware modification! Guaranteed satisfaction! Cat No. 1332

TRS-80 FORTRAN $340 PLUS

As advertized! Supplied on 2 mini-diskettes, and requires a 32K system with one disk drive. Written by Microsoft, designer of Level II and Level III. Includes Fortran compiler, Z-80 macro assembler, text editor, and linking loader. Adds speed and versatility.

SHUGART SA-400 MINIFLOPPY DRIVE $295

Hard and soft sectoring, single density, 35 track. Requires power supply. Cat No. 1154

VERBATIM 5¼" DISKETTES $27 box of 10

Cat No.	Type	Use
1147	Soft sector	TRS-80, Apple
1148	Hard, 10 hole	North Star
1149	Hard, 16 hole	Micropolis

BUY 7, GET 1 FREE!

*Buy 7 of one type, get the eighth of that type free!

Order by type no.

1702A	$3.50
2708	10.00
4116	11.00
21L02-250	1.20
21L02-450	1.00
21L02-650	.65
2114-450	8.00
5203Q	3.25

DATA CASSETTES 10 for $17

Highest quality, leaderless! With protective plastic case. Cat No. 1142

8" DISKS

• Single density
• IBM Compatible

$40 box of 10

Cat No.	Type
1145	32 sector holes, 1 index hole
1146	IBM 32, 3740, 3540, 3770, 3790

California Computer Systems

MEMORY ADD-ON 16K $65

For APPLE, TRS-80, EXIDY

Everything you need! Installs in minutes, no special tools, no soldering! 250 nsec. Cat No. 1156

FREE CATALOG!

New, hot off the press! Top quality, low cost factory fresh IC's, leds, readouts, semis, computer add-ons, boards, test equipment, books, software, PC aids, and more!

Pay by check, COD, Visa, or Mastercharge. Order by phone or mail. Please include phone no. USA add $1.50 for shipping/handling, or $2.50 for air. Foreign add $2.00 for surface, $5.00 for air. COD's add 85c. All items guaranteed satisfaction for 120 days!

19355 BUSINESS CTR DR. - 6L6 NORTHRIDGE, CA 91324

CIRCLE 88

Listing 2 – Tic-Tac-Toe

```
954 REM TIC-TAC-TOE MOVE EVALUATOR IN RADIO SHACK LEVEL I BASIC.
956 REM THE BOARD POSITION IS SORED IN A(Q) THRU A(Q+8) AS
958 REM                X             X
959 REM      A(Q)      X   A(Q+1)    X    A(Q+2)
960 REM     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
961 REM      A(Q+3)    X   A(Q+4)    X    A(Q+5)
962 REM     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
963 REM      A(Q+6)    X   A(Q+7)    X    A(Q+8)
964 REM                X             X
965 REM
968 REM A POSITION UNOCCUPIED CONTAINS A 0.
972 REM A POSITION OCCUPIED BY THE COMPUTER'S MARK CONTAINS A 1.
974 REM A POSITION OCCUPIED BY THE OPPONENT'S MARK CONTAINS A 4.
975 REM M(L), S(L), AND B(L) FROM THE GENERAL ALGORITHM (SEE FIG. 1)
976 REM   ARE STORED IN A(Q+9),A(Q+10), AND A(Q+11), RESPECTIVELY.
978 REM VALUES OF PARAMETERS:
980 REM    S=12
982 REM    M=10 (SEARCHES UNTIL COMPLETION OF GAME)
984 REM    N=0
986 REM    W=100
988 REM
990 REM
992 REM SUBROUTINE 1000 STORES IN A(Q+9) THE NEXT MOVE FOR BOARD
994 REM POSITION A(Q)-A(Q+8) FROM PREVIOUS MOVE A(Q+9). IF A(Q+9)=0,
996 REM FIRST MOVE IS RETURNED. IF THERE ARE NO MORE MOVES, A(Q+9)
998 REM IS RETURNE AS ZERO. NEW BOARD IS STORED IN A(Q+S)-A(Q+S+8)
1000 A(Q+9)=A(Q+9)+1
1010 IF A(Q+9)>9 THEN A(Q+9)=0: RETURN
1019 REM IF POSITION IS OCCUPIED, TRY THE NEXT ONE.
1020 IF A(Q+A(Q+9)-1)<>0 GOTO 1000
1030 FOR I=0 TO 8
1040   A(Q+S+I)=A(Q+I)
1050 NEXT I
1059 REM RECORD THE MOVE.
1060 A(Q+S+A(Q+9)-1)=(Z=1)+4*(Z=-1)
107C RETURN
1986 REM
1988 REM TIC-TAC-TOE VERSION OF GENERAL EVALUATION ALGORITHM
1990 L=0: Z=-1: S=12: M=10: N=0: W=100
2000 L=L+1:Q=S*(L-1)+1: Z=-Z
2010 GOSUB 4000
2020 IF (L<=M)*(O=0) GOTO 2050
2030   GOSUB 3000
2040   GOTO 2150
2050 A(Q+9)=N: A(Q+10)=N: A(Q+11)=-Z*W
2060 GOSUB 1000
2070 IF A(Q+9)=N THEN E=A(Q+11) GOTO 2150
2080 GOSUB 2000
2090 IF L=1 GOTO 2110
2100 IF Z*E>=Z*A(Q-S+11) THEN A(Q+11)=E: GOTO 2150
2110 IF (A(Q+10)<>N)*(Z*E<=Z*A(Q+11)) GOTO 2060
2120 A(Q+11)=E: A(Q+10)=A(Q+9)
2130 IF Z*A(Q+11)<W GOTO 2060
2150 L=L-1: Q=Q-S: Z=-Z: RETURN
2986 REM
2988 REM STATIC MOVE EVALUATOR FOR TIC-TAC-TOE.
2990 REM SUBROUTINE EXAMINES A(Q)-A(Q+8) AND RETURNS:
2992 REM   E=100 IF WINNING POSITION FOR THE COMPUTER.
2994 REM   E=-100 IF WINNING POSITION FOR THE OPPONENT.
2996 REM   E=0   IF DRAW POSITION.
2998 REM   E=-0.5 IF NOT A GAME-OVER POSITION.
3000 GOSUB 4000
3010 IF ABS(V)=100 THEN E=V: RETURN
3020 IF V=8 THEN E=0: RETURN
3030 E=-0.5: RETURN
3988 REM
3990 REM GAME-OVER TESTER FOR TIC-TAC-TOE.
3992 REM SUBROUTINE EXAMINES A(Q)-A(Q+8) AND RETURNS:
3994 REM   O=1, V=100 IF WINNING POSITION FOR COMPUTER.
3996 REM   O=1, V=-100 IF WINNING POSITION FOR OPPONENT
3998 REM   O=1, V=8   IF DRAW POSITION.
3999 REM   O=0, V<8   IF NOT A GAME ENDING POSITION.
4000 RESTORE: V=0
4010 FOR I=1 TO 8
4020   READ A,B,C
4030   T=A(Q+A)+A(Q+B)+A(Q+C)
4040   IF T=3 THEN V=100: O=1: RETURN
4050   IF T=12 THEN V=-100: O=1: RETURN
4060   IF (T=5)+(T=6)+(T=9) THEN V=V+1
4070 NEXT I
4080 O=(V=8)
4090 RETURN
4099 REM THIS STORES ALL 8 COMBINATIONS OF POSITIONS FOR WINNING.
4100 DATA 0,1,2,3,4,5,6,7,8,0,3,6,1,4,7,2,5,8,0,4,8,2,4,6
```


Commit an Original Sin
Byte our apple...
(Software that is!)
A FAST NEW IDEA...
For Apple II Users

How About POSTING 1,000 RECORDS to the GENERAL LEDGER in 7 MINUTES?

It's possible with New Ware's **INTER-ACTIVE** system that has been **SERVICE BUREAU TESTED** and **CPA APPROVED!**

Individual Packages Are:

1. General Ledger
2. Accounts Receivable
3. Accounts Payable
4. Inventory
5. Payroll/Personnel

Order Today: Each Package $120.00*

*Disk with Source Code, add $10.00 (shipped only with packages)
*Disk with Demo Data, add $10.00 (shipped only with packages)

The packages have been designed for Apple II single or dual disk systems with 32K of user memory with or without Applesoft 2 Firmware.

Standard No Charge Features:

- Terminal or printer report selection
- High Speed Journal to Ledger Posting
  - A. Automatic General Ledger Journal Transaction generation
  - B. Automatic Inventory transaction generation
  - C. Automatic Accounts Receivable transaction generation
  - D. Real time accounts receivable application system

402/339-7350

NEW WARE
8525 Park Drive
Omaha, Nebraska 68127

Please Note:
All of our packages are guaranteed to operate within design specifications.

CIRCLE 89

Listing 3 – Kalah

```
938 REM KALAH GAME EVALUATOR IN RADIO SHACK LEVEL I BASIC.
940 REM THE BOARD IS STORED IN A(Q) THRU A(Q+13) AS
942 REM
944 REM                       COMPUTER'S PITS
946 REM      A(Q+12)  A(Q+11)  A(Q+10)  A(Q+9)  A(Q+8)  A(Q+7)
948 REM A(Q+13)
                                                       A(Q+6)
950 REM      A(Q)     A(Q+1)   A(Q+2)   A(Q+3)  A(Q+4)  A(Q+5)
952 REM                       OPPONENT'S PITS
954 REM
956 REM A SIMPLE MOVE IS REPRESENTED BY AN INTEGER 0-5 WITH 0
958 REM   REPRESENTING THE PIT FARTHEST FROM THE PLAYER'S KALAH
960 REM   AND 5 THE PIT NEAREST THE PLAYER'S KALAH.
962 REM A CONTINUATION MOVE IS THE SEQUENCE OFI SIMPLE MOVES CODED
964 REM AS FOLLOWS:
965 REM     (MOVE1)*6(I-1) + (MOVE2)*6(I-2) +...+ (MOVEI)*6°.
966 REM THE LCCATION FOLLOWING THIS CONTAINS 6(I-1) TO INDICATE
968 REM THE NUMBER OF CONTINUATIONS.
970 REM M(L) IS STORED IN A(Q+14),A(Q+15)
972 REM S(L) IS STORED IN A(Q+16),A(Q+17)
974 REM B(L) IS STORED IN A(Q+18)
976 REM VALUE OF PARAMETERS:
978 REM    S=19
980 REM    M IS UNDER EXTERNAL PROGRAM CONTROL
982 REM    N=-1
984 REM    W=100
986 REM
988 REM SUBROUTINE 1000 STORES IN A(Q+14),A(Q+15)THE NEXT MOVE FOR
990 REM BOARD POSITION A(Q)-A(Q+13) FROM PREVIOUS MOVE A(Q+14),
992 REM A(Q+15).  IF A(Q+15)=-1, THE FIRST MOVE IS RETURNED.
994 REM IF THERE ARE NO MORE MOVES, A(Q+14) IS RETURNED AS -1.
996 REM THE RESULTING BOARD POSITION IS STORED IN A(Q+S)-A(Q+S+13).
997 REM
998 REM INCREMENT MOVE AND STORE IN T AND R.
1000 A(Q+14)=A(Q+14)+1: T=A(Q+14): R=A(Q+15)
1009 REM INITIALIZE NEW BOARD.
1010 FOR I=0 TO 13: A(Q+S+I)=A(Q+I): NEXT I
1019 REM IF MOVE AT ONE CONTINUATION EXHAUSTED,COME BACK A LEVEL.
1020 IF (INT(T/6)*6=T)*(R<>1) THEN T=T/6: R=R/6: GOTO 1020
1024 REM TEST FOR LAST MOVE.
1025 IF (T=6)*(R=1) THEN A(Q+14)=-1: RETURN
1030 U=T: V=R
1039 REM PULL OUT SIMPLE MOVE.
1040 X*INT(U/V): U=U-X*V: V=INT(V/6)
1049 REM P IS THE PIT POSITION ON BOARD OF MOVE.
1050 P=7*(Z=1)+X
1059 REM IF PIT IS EMPTY, GO GET ANOTHER MOVE.
1060 IF A(Q+P+S)=0 THEN T=T+1: GOTO 1010
1069 REM MAKE THE MOVE.
1070 D=Q+S: FOR I=P+1 TO P+A(D+P)
1080   J=I-INT(I/14)*14
1090   A(D+J)=A(D+J)+1
1100 NEXT I
1110 I=A(D+P): A(D+P)=0
1119 REM IF A NEW CONTINUATION IF SOUND, GO FORWARD A LEVEL.
1120 IF (J=6+7*(Z=1))*(V=0) THEN T=T*6: R=R*6: U=0: V=1
1129 REM MORE CONTINUATIONS?
1130 IF V>0 GOTO 1040
1139 REM TEST FOR CAPTURE.
1140 IF (A(D+J)<>1)+(P+I>=6+7*(Z=1)) GOTO 1170
1150   A(D+6+7*(Z=1))=A(D+J)+A(D+12-J)+A(D+6+7*(Z=1))
1160   A(D+J)=0: A(D+12-J)=0
1169 REM MOVE COMPLETED, SO RETURN.
1170 A(Q+14)=T:A(Q+15)=R:RETURN
1986 REM
1988 REM GENERAL EVALUATION ALGORITHM (FIG. 1) FOR KALAH.
1990 L=0: Z=-1: S=19: N=-1: W=100
2000 L=L+1: Q=S*(L-1)+1: Z=-Z
2010 GOSUB 4000
2020 IF (L<=M)*(O=0) GOTO 2050
2030   GOSUB 3000
2040   GOTO 2150
2050 A(Q+14)=N:A(Q+15)=1:A(Q+16)=N:A(Q+17)=1:A(Q+18)=-Z*W
2060 GOSUB 1000
2070 IF A(Q+14)=N THEN E=A(Q+18): GOTO 2150
2080 GOSUB 2000
2090 IF L=1 GOTO 2110
2100 IF Z*E>=Z*A(Q-S+18) THEN A(Q+18)=E: GOTO 2150
2110 IF A(Q+16)<>N)*(Z*E<=Z*A(Q+18)) GOTO 2060
2120 A(Q+18)=E: A(Q+16)=A(Q+14): A(Q+17)=A(Q+15)
2130 IF Z*A(Q+18)<W GOTO 2060
2150 L=L-1: Q=Q-S: Z=-Z: RETURN
2982 REM
2984 REM STATIC MOVE EVALUATOR FOR KALAH.
2986 REM  E=100 IF WINNING POSITION FOR THE COMPUTER.
2988 REM  E=-100 IF WINNING POSITION FOR THE OPPONENT.
2987 REM ELSE:
2988 REM  E=((CONTENT OF COMP'S KALAH)-(CONTENTS OF OPP'S KALAH))*
2990 REM     (1 + 1/(19 - MAX CONTENTS OF A KALAH))
2996 REM
2998 REM WINNING POSITION?
3000 GOSUB 4000
3010 E=A(Q+13)-A(Q+6)
3020 IF O=1 THEN E=100*((E>0)-(E<0)): RETURN
3030 F=A(Q+13)*(E>0)+A(Q+6)*(E<0)
3040 E=E*(1+1/(19-F))
3050 RETURN
3992 REM
3994 REM GAME-OVER TESTER FOR KALAH.
3996 REM O=1 IF GAME IS OVER.
3998 REM O=0 IF GAME NOT OVER.
4000 O=0
4009 REM TEST FOR A WINNER.
4010 IF (A(Q+13)>18)+(A(Q+6)>18) THEN O=1: RETURN
4019 REM TEST FOR MOVER'S PITS ALL EMPTY.
4020 J=7*(Z=1)
4030 FOR I=J TO J+5
4040   IF A(Q+I)<>0 THEN RETURN
4050 NEXT I
4059 REM THEY ARE, SO EMPTY OTHER'S PITS INTO KALAH.
4060 FOR I=7-J TO 12-J
4070   A(Q+13-J)=A(Q+13-J)+A(Q+I): A(Q+I)=0
4080 NEXT I
4090 O=1: RETURN
```


PDI

HAVING TROUBLE LEARNING BASIC?

STEP BY STEP is an interactive computer course in BASIC that's easy even for beginners. Program Design has developed a logical, structured approach that really works. At the end of STEP BY STEP, you'll be writing programs using all important BASIC commands.

AVAILABLE FOR TRS-80 LEVELS I & II, PET, AND APPLE II

STEP BY STEP:

- presents material in small steps
- provides guided programming practice in each lesson
- tests your progress after each lesson
- teaches actual program writing, not just terms
- is suitable for anyone from junior high up, regardless of math background

10 lessons with quizzes, plus final test 3 cassettes
80 page Workbook $39.95 plus $1.00 shipping

VISA & Master Charge accepted (include number, exp. date, MC include digits above name)

Department 500
Program Design, Inc., 11 Idar Court, Greenwich, Conn. 06830

CIRCLE 8

MARKET INFORMATION SOFTWARE

Now the first complete Market Information System for TRS-80 users. A must system for anyone investing or considering investing in the stock market.

- DATA BASE MANAGEMENT SYSTEM (DBMS)
  Create, update, edit, display and copy the data base. Separate systems for daily, weekly, or individual stock statistics.
- MARKET INFORMATION SYSTEM (MIS)
  Analyze rallies, declines, and market turnarounds. Includes moving average and price/volume analysis. Also has short-term overbought/oversold indicator, essential for short-term timing.
- COMPLETE DATA FILES (DATA)
  Complete sets of data files. Prices shown for one year's data for daily, weekly, and individual stocks. The larger your data base the more analyses you can perform.

PRICE LIST

SYSTEM	DAILY	WEEKLY	SINGLE
DBMS	14.95	14.95	9.95
MIS	19.95	5.95	5.95
DATA	25.95	15.95	9.95

COMPLETE SYSTEM INCLUDING ONE YEAR'S DATA FOR DAILY, WEEKLY, AND FIVE STOCKS . . . 139.95

Send Check to: Market Information Software
7215 Tod Street
Falls Church, VA 22046

Specify DBMS, MIS, or DATA & SYSTEM TYPE or SINGLE STOCK NAME

Write for more information

TRS-80 16K DISK SYSTEM REQUIRED. NO EXTRA RAM NEEDED FOR LARGER DATA BASE.

CIRCLE 90