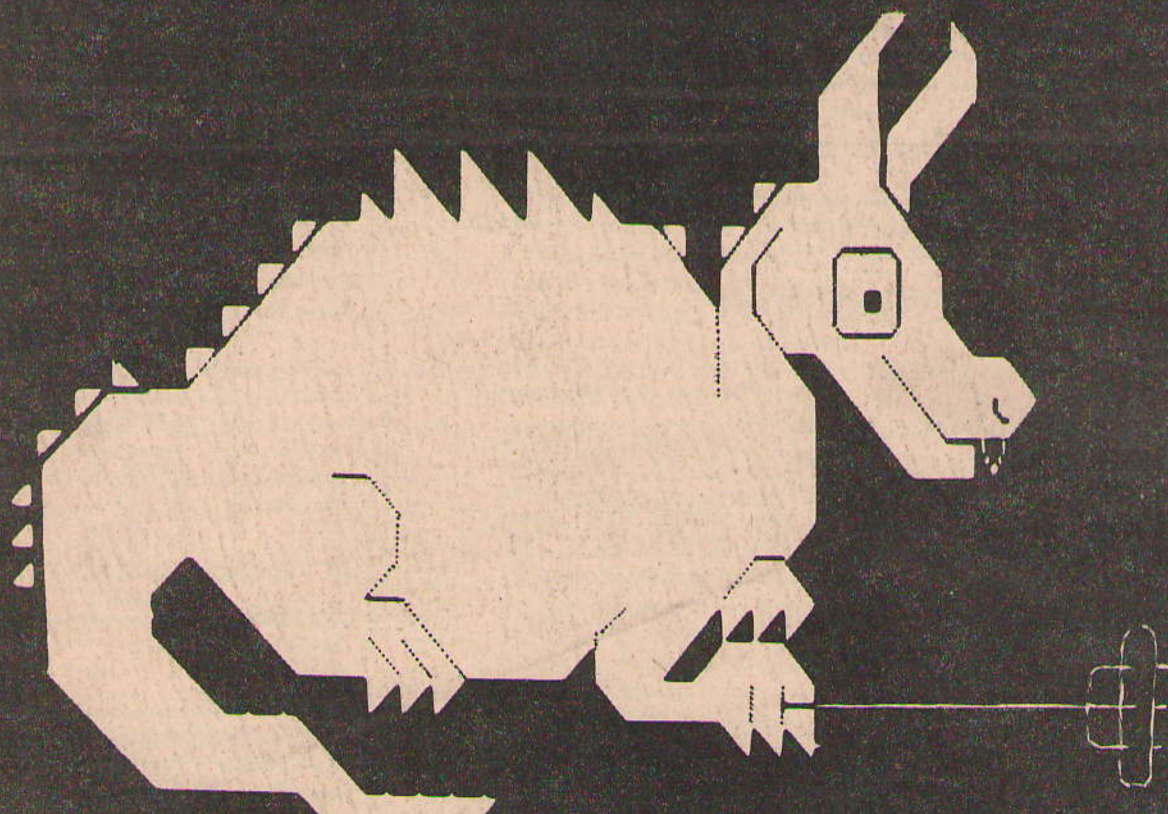
FORMERLY PEOPLE'S COMPUTERS In Canada $2.50 $2.00

Recreational COMPUTING

VOL 8 NO 3 ISSUE 43 NOV-DEC 1979

DRAGON CONTEST RESULTS!



SHOGI: GAMES FOR YOU TO PROGRAM

ATARI SOUNDS • CRYPTARITHMS

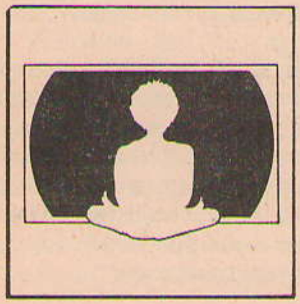
TEXAS INSTRUMENTS GRAPHICS

Recreational COMPUTING

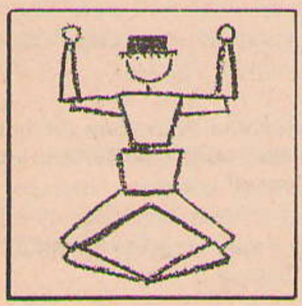
Volume 8 Number 3
Nov-Dec 1979

formerly people's computers

pg. 11



pg. 27



pg. 40

pg. 28

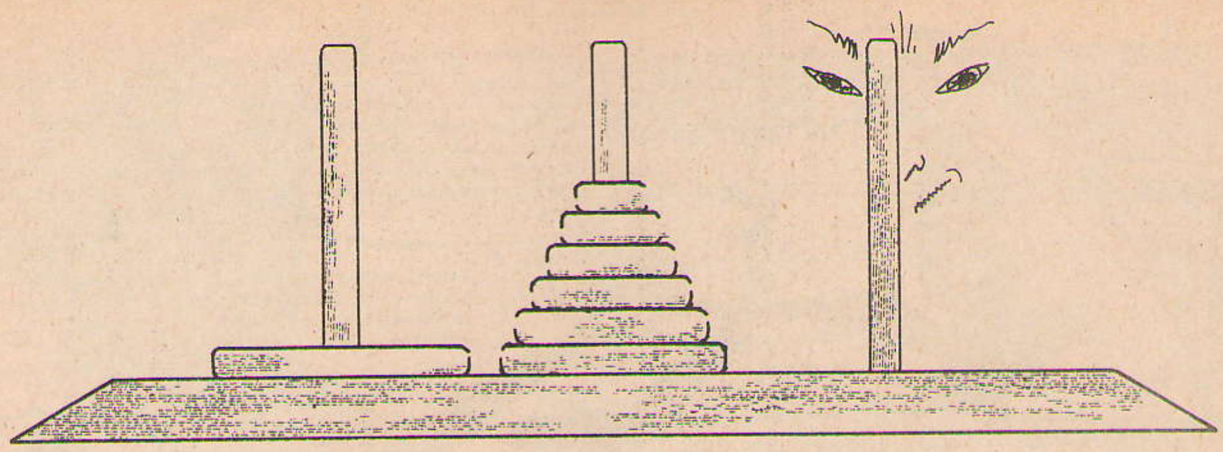
Cover by Ann Miya

Special Features

Articles

Games & Stuff

Departments



TRS-80: TOWER OF HANOI

BY HERBERT L. DERSHEM

Several issues ago we published a couple of program examples using recursion. At that time, we asked if anyone had other home computer applications that were based on recursive procedures. Tower of Hanoi, as presented here, makes excellent use of a recursive technique for implementing the "move" section of the program.

But, beware!! If your micro does not have a recursive BASIC, this program will not work for you as it is written. You will have to alter lines 1000-1100. *– RZ*

This program is a version of the Tower of Hanoi puzzle, also commonly known as Donuts (see "Donuts for Kids" by Ron Sontore in *Calculators/Computers,* November, 1977). The puzzle consists of N doughnut shaped discs, of different diameters, and three vertical towers on which the discs can be placed. The problem begins with all the discs on the left-most of the three towers, called Tower 1. The larger discs are at the bottom of the stack.

The object is to move all of the discs from Tower 1 to Tower 2, the middle of the three towers. There are three rules to follow in moving the discs:

1) Only one disc can be moved at a time.
2) A disc is moved by removing it from the top of the stack of discs of one tower and placing it on the top of a stack of discs of another tower.
3) No disc may ever be placed on top of a smaller disc.

The program given here is written in Level II BASIC for the Radio Shack TRS-80. The user has the options of solving the puzzle himself or watching the computer solve it. In either case, the user has the choice of the number of discs to be used, up to a maximum of 15. The program also provides a time estimate for how long the solution will take. This is helpful in the case of 15 discs where at least 32,767 moves are required. When the computer solves the puzzle, the user has the option of controlling the length of time between moves. This feature is helpful for demonstration purposes.

The method used by the computer is recursive, which means that the routine to determine the moves calls upon itself. The basic principle here is that if you have a routine to successfully move n-1 discs from Tower i to Tower j, for any i and j, then you can move n discs from Tower i to Tower j by the following procedure:

1) Move n-1 discs from Tower i to Tower k.
2) Move 1 disc from Tower i to Tower j.
3) Move n-1 discs from Tower k to Tower j.

where k is the number of the Tower which is not Tower i or Tower j. In other words, $k = 6 - i - j$. This recursive procedure is shown in statements 1000-1100.

This program has been used with students at both the elementary and the secondary level. The puzzle is explained by showing the students the computer solution with three discs. Usually, about a three second delay is requested between moves so that an explanation of each move can be given. Following this, the students are asked to solve the puzzle with four discs and compare their number of moves to what the computer reports as optimum. After several tries or after they do obtain the optimum number of moves, they watch the computer solve the puzzle for four discs, and then go on to try it for five. After the optimum effort for five discs is found, the students are encouraged to conjecture on the effort needed for six or more. For a larger number of discs, running the program at least to the point where the time estimate is produced is also interesting.

TOWER LISTING

```
10   REM TOWER OF HANOI PUZZLE IN LEVEL II BASIC
20   KT=0: PRINT "TOWER OF HANOI"
30   INPUT "DO YOU WANT INSTRUCTIONS"; R$
40   GOSUB 4000
50   ON R GOTO 500,60,30
60   INPUT "HOW MANY DISCS DO YOU WANT"; ND
70   R=ND: LO=1: H1=15: GOSUB 3000
80   IF B=1 THEN 60
90   DIM N(3),D(3,ND),E1(ND),F1(ND),D1(ND)
100  INPUT "DO YOU WANT TO MAKE THE MOVES YOURSELF";R$
110  GOSUB 4000
120  ON R GOTO 300,130,100
130  INPUT "HOW MANY SECONDS DELAY DO YOU WANT BETWEEN MOVES";NS
140  R=NS: LO=0: H1=10: GOSUB 3020
150  IF B=1 THEN 130
160  T=(2↑ND-1)*(NS+.57)
170  GOSUB 6000
180  D=ND: E=1: F=2
190  GOSUB 5000
200  GOSUB 990
210  FOR I=1 TO 100: PRINT 896, "(APPLAUSE! APPLAUSE!)";
220  FOR J=1 TO 100: NEXT J: PRINT 896,CHRS(30);
230  FOR J=1 TO 100: NEXT J: NEXT I
240  STOP
290  REM USER MAKES MOVES.
300  T=(2↑ND-1)*5.57
310  PRINT "ASSUMING IT TAKES YOU 5 SECONDS PER MOVE"
320  GOSUB 6000
330  D=ND: E=1: F=2
340  GOSUB 5000
350  PRINT@768,CHR$(30);: PRINT@768,"ENTER MOVE";KT+1;": FROM"
360  INPUT E
370  R=E: LO=1: H1=3: GOSUB 3000
380  IF B=1 THEN 350
390  PRINT@768,CHR$(30);:PRINT @832, CHR&(30);
400  PRINT@768,"ENTER YOUR MOVE";KT+1;": FROM";E;"TO";
410  INPUT F
420  R=F: GOSUB 3000
430  IF B=1 THEN 400 ELSE PRINT@832,CHR$(30);
440  (E=F) OR (N(E)<=0) OR (N(F)<>0 AND D(E,N(E))>D(F,N(F))) THEN PRINT@832,
         "ILLEGAL MOVE";CHR$(30);: GOTO 350
450  GOSUB 2000
460  IF N(2)<ND THEN 350
470  IF KT>2↑ND-1 THEN PRINT@896,"YOU DID IT, BUT YOU COULD HAVE DONE IT IN";
         2↑ND-1; "MOVES.";
480  IF KT=2↑ND-1 THEN PRINT@896,"CONGRATULATIONS! YOU DID IT IN THE LEAST
         POSSIBLE MOVES.";
490  FOR I=1 TO 10000: NEXT I: STOP
500  REM PRINT INSTRUCTIONS
510  PRINT "YOU ARE GIVEN THREE TOWERS ON WHICH YOU CAN PLACE DISCS WITH"
520  PRINT "HOLES IN THE CENTER. THE DISCS ARE ALL ON TOWER 1 TO START"
530  PRINT "WITH THE LARGEST ON THE BOTTOM AND THE SMALLEST ON TOP.THE GOAL"
540  PRINT "IS TO MOVE THE DISCS ONE AT A TIME SO THAT ALL DISCS END UP ON"
550  PRINT "TOWER 2. THE ONLY RESTRICTION IS THAT A LARGER DISC MAY NEVER"
560  PRINT "BE PLACED ON A SMALLER ONE. YOU HAVE THE OPTION OF"
570  PRINT "SOLVING THE PUZZLE YOURSELF OR ASKING THE COMPUTER TO SOLVE IT."
580  PRINT "GOOD LUCK!"
590  GOTO 60
970  REM SUBROUTINE TO RECURSIVELY MOVE D DISCS FROM TOWER E TO TOWER F.
980  REM FIRST ENTRY IS TO LINE 990.  SUBSEQUENT CALLS ARE TO LINE 1000.
990  S=1
1000 IF D=1 THEN GOSUB 2000
1010 IF D=1 THEN 1090 ELSE G=6-E-F
1020 E1(S)=E: F1(S)=F: D1(S)=D
1030 S=S+1: D=D-1: F=G
1040 GOSUB 1000
1050 E=E1(S): F=F1(S): D=D1(S)
1060 GOSUB 2000
1070 S=S+1: D=D-1: E=6-E-F
1080 GOSUB 1000
1090 S=S-1
1100 RETURN
2000 REM MOVE A DISC FROM TOWER E TO TOWER F.
2010 KT=KT+1: B=ND+10: PRINT@0,STR$(KT);".MOVE FROM";E;"TO";F;
2020 N(E)=N(E)-1:N(F)=N(F)+1
2030 SZ=D(E,N(E)+1)
2040 D(F,N(F))=SZ
2050 XE=40*E-20
2060 XF=40*F-20
2070 FOR J=-SZ-1 TO SZ+1
2080 IF J<>0 THEN RESET (XE+J,B-N(E)): SET(XF+J,B-N(F)+1)
2090 NEXT J
2100 FOR J=1 TO 500*NS: NEXT J
2110 RETURN
3000 REM CHECK RESPONSE R TO SEE IF IT IS BETWEEN LO AND H1.
3010 IF R<>INT(R) THEN PRINT "RESPONSE MUST BE AN INTEGER": B=1:RETURN
3020 IF R<LO THEN PRINT "RESPONSE MUST BE NO LESS THAN";LO: B=1:RETURN
3030 IF R>H1 THEN PRINT "RESPONSE MUST BE NO MORE THAN";H1: B=1:RETURN
3040 B=0: RETURN
4000 REM CHECK RESPONSE R$ FOR YES OR NO.
4010 IF R$="YES" THEN R=1: RETURN
4020 IF R$="NO" THEN R=2: RETURN
4030 PRINT "PLEASE ENTER YES OR NO.": R=3: RETURN
5000 REM DRAW TOWERS
5010 CLS
5020 L=INT((ND+10)/2)*64-64
5030 FOR I=1 TO 3
5040 PRINT@L+20*I-12,"TOWER";I;
5050 NEXT I
5060 PRINT@L+150,"TOWER OF HANOI";
5070 FOR I=1 TO ND
5080 D(1,I)=ND+1-I
5090 NEXT I
5100 N(1)=ND: N(2)=0:N(3)=0
5110 B=ND+10
5120 FOR I=6 TO B
5130 SET(20,I): SET(60,I): SET(100,I)
5140 NEXT I
5150 FOR I=B TO B-ND+1 STEP-1
5160 W=ND-B+I-1
5170 FOR J=20-W TO 20+W
5180 SET(J,I)
5190 NEXT J
5200 NEXT I
5210 RETURN
6000 REM PRINT TIME ESTIMATE
6010 PRINT "TIME ESTIMATE FOR THIS PUZZLE:";
6020 D=0: H=0: M=0
6030 IF T>=86400 THEN D=INT(T/86400): PRINT D; "DAYS";
6040 T=T-D*86400
6050 IF T>=3600 THEN H=INT(T/3600): PRINT H; "HOURS";
6060 T=T-H*3600
6070 IF T>=60 THEN M=INT(T/60): PRINT M;"MINUTES";
6080 T=T-M*60
6090 PRINT T; "SECONDS".
6100 FOR I=1 TO 1500: NEXT I
6110 RETURN
```