PRICE $ 2.00

THE 80 NOTEBOOK

APRIL 1980 ISSUE #1

NOTE: The term "TRS-80" is a registered trademark of Radio Shack, a Division of Tandy Corporation. THE 80 NOTEBOOK is not affiliated with Radio Shack or Tandy Corporation in any way.

WELCOME TO THE 80 NOTEBOOK

Let me take this opportunity to introduce myself to you. My name is Michael Clark and I am the editor of THE 80 NOTEBOOK. Our offices are located in rural Pennsylvania and can be reached by mail at R. D. #3, Nazareth, PA 18064 or by phone at 215-759-6873.

While our publication is a small one, we will try our best to bring you the finest articles and programs possible for the TRS-80. Because we do not carry advertising from TRS-80 vendors, we can devote the pages of THE NOTEBOOK to things of interest to you, the reader. This is not to say that we intend to ignore the wide variety of hardware and software products currently available for the TRS-80 or any such products marketed in the future. We would prefer to analyze new products and objectively review them in future articles in THE NOTEBOOK to help you, the TRS-80 user, get only the best available. In the same spirit, our publication will give you, each month, the best we have to offer in articles covering a wide range of programming interests and our articles will always carry complete, well-documented source listings of all programs discussed in THE NOTEBOOK.

Some of the various programming topics to be covered in future issues include: games, computer assisted instruction, adventure scenerios, word processing, new peripheral hardware, computer simulation techniques, operating systems and languages, business software, data base management,BASIC and Assembly Language programming lessons, and artificial intelligence, just to name a few.

For our first issue, we have chosen to entertain you with some games. Our feature article offers you a detailed look at a computerized version of the popular game - MONOPOLY (Monopoly is a trademark of Parker Brothers Games). This version of Monopoly is not just a demonstration of the basic workings of the game but is a program which will enable your TRS-80 to actively play the game as one of its players, both obey and utilize all the capabilities provided in the rules and act as banker, allowing up to 8 human players to challenge the computer to a game of Monopoly. The program provides all the visual aids needed to play the game without having the box version of Monopoly present. The level of detail provided by this article and its program is not often found in other magazines' articles!

Not all our articles come from our staff. If you have an interesting or unusual use for your TRS-80 whether it be a game, simulation or whatever, write an article about your topic and submit it to THE NOTEBOOK for our consideration. Be sure you include all source listings and documentation so our readers can both understand and enjoy your idea. If your article is chosen for publication, we will pay you $15 for each typeset page your article takes in our publication.

To aid in typing BASIC source listings in your articles, we have devised the following list of special character symbol replacements: .GT. means greater than, .LT. means less than, .EQ. means equal to, .GE. means greater than or equal to, .LE. means less than or equal to, .NE. means not equal to. These symbol replacements make "IF" statements more understandable and easier to type on a conventional typewriter. All the source listings we print will carry these symbol replacements. When entering source programs from THE 80 NOTEBOOK into your TRS-80, be sure to use the proper symbols for each of the six character typed symbol replacements.

With all of this available to you, we hope you will agree that THE 80 NOTEBOOK is the TRS-80 publication for you. We hope you enjoy this first issue and every issue for many years to come.

••

NOTICE: Any individual wishing to trade or sell TRS-80 hardware can advertise free in our classified section.

Anyone wishing to comment on new products should send a letter to the editor for publication or submit an article detailing the use of a new product.

Any TRS-80 clubs may submit club news to be published free of charge.

••

A COMPUTERIZED VERSION OF PARKER BROTHERS' MONOPOLY

This month we are featuring our version of Parker Brothers' popular game Monopoly. This program allows the TRS-80 to play the game along with up to 8 human players. Virtually every phase of the game and the instructions for the game are simulated by this program along with a game strategy algorithm used by the program in its playing of the game. The program is designed for a Level 2 16K TRS-80. Here is an outline of the program's various segments and their functions:

INITIALIZATION SECTION - Lines 5 thru 46:

This section of the program makes all the preparations for the start of the game. This is the only area of the program which departs from the instructions for the box version of the game. For one thing, the program only plays the (regular) long version of the game. It always acts as the banker for the game. In place of using tokens, each player is identified by a number from 1 to 8 with the TRS-80 being identified by the number 9. Prior to the start of the program, the human players must choose the order in which they will take their turns at play with that order being the same as the token number of each player. This information is conveyed to the program when the program asks how many players are challenging it and what their names are. The players' names are stored in array A1$ with the number of players stored in variable "P".

Some of the other array and variable fields initialized at this time include:

"A" - A 40 element array representing each square on the board. This array will show the token numbers of the owners of the various properties on the board.

A7 - A 40 element array showing the number of houses (0 to 4) or hotels (5) on each property on the board. For the start of the game. both "A" and A7 are initialized to 0 - unoccupied.

A8 - A 16 element array containing the Community Chest card deck after shuffling that deck.

A9 - A 16 element array containing the Chance card deck after shuffling that deck.

Within the initialization section, GOSUBs are performed to shuffle the Chance and Community Chest card decks (GOSUB 3000 and GOSUB 3100).

For each player, array A2 is initialized to 1500 showing that players' cash balance; array B is initialized to 1 showing which square on the board that player is sitting on.

Array A6 is initialized to 0 showing that no player has a "get out of jail free" card. A6 will be set to 1 if a Community Chest "get out of jail free" card is drawn by a player, 2 if a Chance "get out of jail free" card is drawn by a player, and 3 if both "get out of jail free" cards are drawn by a player. When a "get out of jail free" card is used by a player, A6 will be reset accordingly.

Some of the other variables used by the program for game control or in its role as a player include:

TC - Set to 1 if the Community Chest "get out of jail free" card is drawn and retained by a player.

TG - Set to 1 if the Chance "get out of jail free" card is drawn and retained by a player.

Finally, the initialization section displays the "GO" square where every player's token starts at the beginning of the game. This is done through a GOSUB to line 2000. This will display the following screen if, for example, Tom and Bob are playing against the computer:

```
PLAYER NONE CASH 0
START
SQUARE 1
PROPERTY - GO
OWNER - NONE
TOKENS - TOM BOB TRS-80
```

PLAYER ACTION SECTION - Lines 48 thru 501:

In this section of the program, an involved series of instructions is performed to simulate the actions of both the human and computer players.

The variable I will indicate the number of the player being simulated in the loop. The variable S indicates the number of doubles a player has rolled within one turn. If S = 3, the player is sent to jail per Monopoly's normal rules.

If sent to jail during a turn, a player must stay there until his next turn; at which time the player may use a "get out of jail free" card or pay the $50 fine to get out of jail. If the player elects to stay in jail, he must roll a set of doubles or pay the $50 fine within three turns after being sent to jail in order to get out of jail. After three turns in jail and unable to pay the fine, a player is forced to quit!

Variable B(I) indicates the square a player's token is on at the start of a turn. If B(I)=0, that player is no longer in the game.

By using a random number, variables X and Y are set to the rolls of the dice for a player's turn. Tha location of a player's token is displayed followed by a display of token's position after moving the roll of the dice. If the move places the token on or past the GO square, the player collects 200 dollars. If the player lands on JUST VISITING or FREE PARKING, nothing happens. If the player lands on INCOME TAX or LUXURY TAX, he must pay 200 dollars or 75 dollars respectively.

If a player must pay a certain amount to any other player or the bank during his turn and has no cash, he is given the opportunity to mortgage or sell houses and hotels back to the bank for the cash specified by the normal Monopoly rules. If the resulting cash balance is insufficient to pay the player's debt, the player is forced to quit.

If the player lands on either the COMMUNITY CHEST or CHANCE squares, the program simulates the drawing of a card from the appropriate deck and acting on the information on the card. Lines 400 to 490 resolve any effects of the card draw. Among these effects are: the change in cash levels in variable BG, the square your token is relocated to in variable BH, the jailing status by variable BI, the giving or collecting of cash from or to all other players in variable BJ and the rent multiplication factor in variable BW.

If you land on a square where rent is owed to another player, the rent is computed and distributed. If the property is unowned, the player has the option to buy it. The property remains unowned if not purchased by that player at that time. This differs from the auction and private buying and selling of properties allowed in the normal Monopoly rules.

If the player's roll of the dice was a double, his turn continues with another roll of the dice unless three doubles in one turn has sent him to jail.

PLAN TO GET OUT OF JAIL - Lines 1100 thru 1199:

This section of the program allows the player who has landed in jail to use a GET OUT OF JAIL FREE card or pay the $50 fine to get out of jail before making his first move within a turn. Logic is provided to enable the computer to make these decisions based on cash and free card availability. So long as it has enough cash, the computer will always elect to get out of jail on his next turn in order to continue buying properties and collect $200 GO money.

DISPLAY A TITLE DEED FOR A PROPERTY - Lines 2000 thru 2365:

This section of the program serves as a subroutine used to display all the information available concerning the property or square on which a player's token is located. It is called by a number of other sections of the program both at the start of and during the playing of the game.

Once the screen is cleared, the name of the player whose turn it is , is printed along with his or her cash balance. Then, a section supplied message is printed along with the number of the square on which that player's token rests. The section supplied message is a parameter passed to the subroutine in the variable M9$. The square number is passed in AZ and the pointer to the name of the player is passed in M8.

After this, another subroutine is called through a GOSUB to 3205. This call serves to look up all the statistics about the property or square pointed to by AZ. Upon return, if AF$ is blank, then the square in question is not a property and a further determination must be made to assign the name of the square in AF$. This determination would resolve "GO", "COMMUNITY CHEST", "INCOME TAX", "CHANCE", "JAIL", "FREE PARKING", "GO TO JAIL" or "LUXURY TAX".

At this time, the property name AF$ is printed along with the name of the owner of the property with "NONE" meaning not owned and "TRS-80" meaning owned by the computer. If the property is currently mortgaged, the routine informs you of this.

If the square is the JAIL, the routine next prints who is in JAIL followed by who is JUST VISITING, otherwise the routine prints the name of the color group to which that property belongs along with the tokens sitting on that property.

The color group name is contained in FZ$ after returning from a GOSUB to line 6000 which determines the name of the color group to which a property belongs and the owner of the color, if any.

If the square in question is a property, the price, mortgage value, rent and improvement cost, if any, is printed. If the property is a utility, the rent is expressed in a multiple of the throw of the dice. If the property is a railroad, the rent is expressed as a function of the number of railroads owned by one player. If the property is a member of a color group, the rent is expressed as a function of the number of houses or a hotel on that property.

If the square is a color group property, the number of houses or a hotel is printed along with the cost of buying additional houses or trading in four houses for a hotel on that property. Finally, the screen pauses for 2 seconds before the routine returns to the line in the program which called this routine.

Here are some examples of what you see on the screen for various tokens on different types of properties:

EXAMPLE 1:

```
PLAYER TOM CASH 1500
NEW LOCATION
SQUARE 3
PROPERTY - COMMUNITY CHEST
OWNER - NONE
TOKENS - TOM BOB TRS-80
```

EXAMPLE 2:

```
PLAYER TOM CASH 1250
NEW LOCATION
SQUARE 11
PROPERTY - JAIL
OWNER - NONE
IN JAIL - BOB
JUST VISITING
TOKENS - TOM TRS-80
```

EXAMPLE 3:

```
PLAYER TOM CASH 925
NEW LOCATION
SQUARE 6
PROPERTY - READING RR
OWNER - BOB
CURRENTLY MORTGAGED
COLOR GROUP - RAILROADS
TOKENS - TOM
PRICE - 200
MORTGAGE VALUE - 100
RENT - 25
IF 2 RR OWNED 50
IF 3 RR OWNED 100
IF 4 RR OWNED 200
```

EXAMPLE 4:

```
PLAYER TOM CASH 1425
```

```
NEW LOCATION
SQUARE 13
PROPERTY - ELECTRIC CO
OWNER - NONE
COLOR GROUP - UTILITIES
TOKENS - TOM
PRICE - 150
MORTGAGE VALUE - 75
IF 1 UTILITY OWNED, RENT = 4X DICE THROWN
IF BOTH UTILITIES OWNED, RENT = 10X DICE THROWN
```

EXAMPLE 5:

```
PLAYER TOM CASH 1300
NEW LOCATION
SQUARE 40
PROPERTY - BOARDWALK
OWNER - BOB
COLOR GROUP - BLUE
TOKENS - TOM
PRICE - 400
MORTGAGE VALUE - 200
RENT - 50
WITH 1 HOUSE - 200
WITH 2 HOUSES - 600
WITH 3 HOUSES - 1400
WITH 4 HOUSES - 1700
WITH HOTEL - 2000
4 HOUSES ON PROPERTY
IMPROVEMENT COST - 200
```

In all of these examples, Tom, Bob and the computer were used as the players involved with the various properties. The term "NEW LOCATION" indicates that the player just landed on the square after a move. As you can see, all the information normally provided by the box game is supplied by the computer. Before a player makes a move, the "OLD LOCATION" they were at is first displayed in the same manner. This display function is also invoked by other player situations where the message "NEW LOCATION" is replaced by an appropriate explanation.

PAY INCOME TAX - Lines 2400 thru 2499:

This section of the program allows a player to produce the 200 dollars for INCOME TAX when a player lands on the INCOME TAX square. If needed, a player may mortgage properties to gain the needed cash.

PAY LUXURY TAX - Lines 2600 thru 2699:

This section of the program allows a player to produce the 75 dollars for LUXURY TAX when a player lands on the LUXURY TAX square. If needed, a player may mortgage properties to gain the needed cash.

MORTGAGE PROPERTIES FOR CASH - Lines 2800 thru 2899 and lines 7100 thru 7199:

This section of the program allows a player to raise additional cash for the payment of a debt by selling houses and hotels he owns back to the bank for half their improved value and to mortgage unimproved properties with the bank for half their purchase price. Lines 7100 thru 7199 contain the logic needed for the computer to make the decisions needed to mortgage its properties and sell houses back to the bank for cash.

SHUFFLE THE COMMUNITY CHEST DECK - Lines 3000 thru 3099:

This section of the program allows the computer to randomize the sequence of the cards in the COMMUNITY CHEST deck. It also keeps track of the whereabouts of the GET OUT OF JAIL FREE card.

SHUFFLE THE CHANCE DECK - Lines 3100 thru 3199:

This section of the program allows the computer to randomize the sequence of the cards in the CHANCE deck. It also keeps track of the GET OUT OF JAIL FREE card.

ASSIGN PROPERTY STATISTICS - Lines 3200 thru 3499:

This section of the program looks up the name, price, rents, mortgage value and improvement value of the square number passed in variable AZ.

COMPUTE RENT TO PAY - Lines 3600 thru 3999:

This section of the program computes the rent a player must pay depending on the number of improvements existing on an owned property he has landed on. This rental is stored in variable AP but must later be multiplied by the rent factor BW which is affected by a CHANCE card.

DRAW A COMMUNITY CHEST CARD - Lines 4000 thru 4299:

This section of the program simulates the drawing of a card from the COMMUNITY CHEST deck and sets up the needed variables for taking the actions dictated.

DRAW A CHANCE CARD - Lines 4300 thru 4799:

This section of the program simulates the drawing of a card from the CHANCE deck and sets up the needed variables for taking the actions dictated.

PAY RENT TO A PLAYER - Lines 5000 thru 5099:

This section of the program transfers cash funds between players in the paying of a rental.

DETERMINE COLOR GROUP - Lines 6000 thru 6399:

This section of the program looks up the color group name and the owner's player number for square number AZ.

UNMORTGAGE PROPERTIES, BUY HOUSES AND HOTELS - Lines 6500 thru 6699 and lines 7200 thru 7299:

At the end of a player's turn, this section of the program allows a player to unmortgage any of his properties and to buy houses and hotels for improved properties for which he owns all the properties in that color group. Lines 7200 thru 7299 provides the logic needed to allow the computer to unmortgage and improve its properties.

CONCLUSIONS:

As you can see, most aspects of the normal Monopoly game rules are simulated in this game program. The program normally ends when all but one player has gone broke and been forced to quit but can be interrupted by the break key at any time. We hope you will enjoy challenging the computer to this old and well-loved box game. You must be familiar with the Monopoly game and its rules before using this program or trying to enhance its capabilities.

To save inputting the program yourself, a CLOAD tested cassette copy of the program is available only from THE 80 NOTEBOOK for the modest price of $4.95 postpaid. One final note: because of the size of this program, you must remove the REM statements in order to fit into a 16K machine!

```
1 REM MONOPOLY, COPYRIGHT 1980, THE 80 NOTEBOOK
5 REM  INITIALIZATION SECTION
7 CLEAR 500:CLS:PRINT"MONOPOLY"
10 RANDOM:DEFINTA-Z:DIMA(40)
12 DIM A7(40),A8(16),A9(16)
20 T=1500:L=0:P=0:Z=1:G=0:Q=0:F=0:HA=0:HB=0
25 TC=0:TG=0:GOSUB3000
30 FORI=1TO40:A(I)=0:NEXTI
35 FORI=1TO40:A7(I)=0:NEXTI:GOSUB3100:B2=0
40 INPUT"#PLAYERS";P:IFP<1ORP>8THEN40
42 DIM B(9),A1$(9),A2(9),A6(9):A1$(0)="NONE":A2(0)=0
43 FORI=0TO9:B(I)=0:A1$(I)=" ":A2(I)=0:A6(I)=0:NEXTI
44 FORI=1TOP:B(I)=1:A6(I)=0:A2(I)=1500:PRINT"PLAYER";I
45 INPUT"YOUR NAME";A1$(I):NEXTI:M8=0:A1$(9)="TRS-80"
46 M9$="START":A2(9)=1500:A6(0)=0:A6(9)=0:B(9)=1
47 A1$(0)="NONE":A7=1:GOSUB2000
48 REM PLAYER ACTION SECTION
50 FORI=1TO9
51 S=0
52 IFI=9ANDB(I)>0ANDP=FTHENPRINT"I WIN!":END
53 BW=1:IFB(I)>0ANDP=FTHENPRINTA1$(I);" WINS!":END
55 IFB(I)=0THEN500
60 X=INT(RND(0)*6):IFX=0THENX=1
65 GOSUB2355
70 Y=INT(RND(0)*6):IFY=0THENY=1
80 A7=B(I):M8=I:M9$="OLD LOCATION":GOSUB 2000
81 PRINT"DICE ROLL";X;Y:M9$="NEW LOCATION":GOSUB 2355
82 IFX=YANDB(I)<41THENS=S+1
83 IFS=3 THEN B(I)=43:PRINT"3 DOUBLES - GO TO JAIL":A7=11:GOSUB 2000:GOTO 500
85 IFB(I)>40ANDX=YTHENB(I)=11:PRINT"OUT OF JAIL"
87 IFB(I)>40THENGOSUB1100
88 IFB(I)=0 THEN 500
90 W=X+Y
95 IFB(I)>40THENB(I)=B(I)-1:GOTO500
97 IFB(I)+W>40 THEN A2(I)=A2(I)+200
100 B(I)=B(I)+W:IFB(I)>40THENB(I)=B(I)-40:PRINT"COLLECT $200"
103 A7=B(I):GOSUB2000:ER=0:BV=0
105 IFB(I)=1THENPRINT"ON GO":GOTO490
110 M=B(I)
115 IFM=31THENB(I)=43:PRINT"GO TO JAIL":GOTO500
116 IFM=8 OR M=23 OR M=37 THEN GOSUB 4200
117 IFM=5 THEN GOSUB 2400
118 IFM=39 THEN GOSUB 2600
```

```
119 IFM=3 OR M=18 OR M=34 THEN GOSUB 4000
120 IFB(I)=0 THEN 500
122 IFM=21THENPRINT"FREE PARKING":GOTO490
124 IFM=5 OR M=39 THEN 490
126 IFM=11THENPRINT"JUST VISITING":GOTO490
128 IFM=3ORM=18ORM=34THENPRINT"COMMUNITY CHEST":GOTO400
130 IFM=8ORM=23ORM=37THENPRINT"CHANCE":GOTO400
132 IFA(M)>0ANDA(M)<>IANDA(M)<11THENPRINT"PAY RENT TO PLAYER";A(M);"IF PROPERTY
NOT MORTGAGED"
133 IFA(M)>0 AND A(M)<>I AND A(M)<100 THEN GOSUB 5000
134 IFB(I)=0THEN500ELSEA7=B(I):GOSUB3205
135 IFA(M)>0THEN490
136 MJ=AG:M2=I:IFI<9 THEN 140
138 IFA2(I)<AG THEN 490
139 GOTO151
140 V$=" ":INPUT"BUYING PROPERTY";V$
150 IFV$<>"Y"THEN490
151 IFA2(I)<AG THEN GOSUB 2800
152 IFB(I)=0 THEN 500
153 A7=B(I):GOSUB3205
154 PRINTA1$(I);" OWNS IT!"
155 A2(I)=A2(I)-AG:A(M)=I:GOTO490
400 REM RESOLVE ACTIONS OF CHANCE AND COMMUNITY CHEST
402 M2=I:GOSUB 2355
405 IFBJ>0 THEN 450
409 MJ=A2(I)+BG:IFMJ<0THENMJ=MJ*-1
410 IFA2(I)+BG<0 THEN GOSUB 2800
412 IFB(I)=0THEN500ELSEA7=B(I):GOSUB3205
413 A2(I)=A2(I)+BG
415 IFBI=1 THEN B(I)=43:GOTO 500
418 IFBH>0 AND BH<B(I) THEN A2(I)=A2(I)+200
420 IFBH>0 THEN B(I)=BH
425 IFBI=2 THEN A6(I)=A6(I)+1:TC=1
430 IFBI=3 THEN A6(I)=A6(I)+10:TG=1
440 IFBH>0 THEN 103
445 GOTO 490
450 IFBJ=2 THEN 475
452 FORM2=1TO9:IFB(M2)=0 THEN 495
455 IFI=M2 THEN 495
465 IFA2(M2)<BG THEN MJ=BG:GOSUB 2800
467 IFB(M2)=0 THEN 471
470 A2(I)=A2(I)+BG:A2(M2)=A2(M2)-BG
471 NEXTM2
```

```
472 GOTO 490
475 IFA2(I)>(P-F)*BG THEN MJ=(P-F)*BG:GOSUB 2000
476 IFB(I)=0 THEN 500
477 FORM8=1TO9:IFB(M8)=0 OR B(M8)=I THEN 481
480 A2(I)=A2(I)-BG:A2(M8)=A2(M8)+BG
481 NEXTM8
482 GOTO 490
490 IFX=YANDS>0THENPRINT"ROLL AGAIN":GOTO60
495 IFA2(I)>30 THEN GOSUB 6500
500 NEXTI
501 GOTO 50
1100 REM PLAN TO GET OUT OF JAIL
1102 M2=I:MJ=50
1105 IFA6(I)=0 THEN 1145
1107 R1=0:R2=0
1110 IFA6(I)>9 THEN R2=A6(I)-10:R1=1 ELSE R2=1
1112 IFI=9 AND R2>0 THEN R3=1:GOTO 1130
1113 IFI=9 AND R1>0 THEN R3=2:GOTO 1130
1115 IFR1=1 THEN PRINT"YOU HAVE A CHANCE GET OUT OF JAIL FREE CARD"
1120 IFR2=1 THEN PRINT"YOU HAVE A COMMUNITY CHEST GET OUT OF JAIL FREE CARD"
1125 INPUT"ENTER 1 = USE COMMUNITY CHEST CARD, 2 = USE CHANCE CARD, 3 = USE NO C
ARD";R3
1126 IFR3=3 THEN 1145
1127 IFR3=1 AND R2=0 THEN 1125
1128 IFR3=2 AND R1=0 THEN 1125
1130 B(I)=11:IFR3=1 THEN TC=0:A6(I)=A6(I)-1
1135 IFR3=2 THEN TG=0:A6(I)=A6(I)-10
1137 PRINT"USED GET OUT OF JAIL FREE CARD!"
1140 RETURN
1145 IFB(I)=41 THEN 1160
1147 IFI=9 AND A2(I)>50 THEN 1160
1148 IFI=9 THEN RETURN
1150 INPUT"PAYING THE $50 FINE (Y/N)";V$
1152 IFV$<>"Y" THEN RETURN
1160 IFA2(I)<50THENGOSUB2000
1165 IFB(I)=0 THEN RETURN
1168 PRINT"PAID FINE!"
1170 B(I)=11:A2(I)=A2(I)-50:RETURN
2000 REM DISPLAY TITLE DEED FOR PROPERTY (AZ) ROUTINE
2002 IFAZ>40 THEN AZ=11
2003 Z=B(9)
2005 CLS:PRINT"PLAYER ";A1$(M8);" CASH ";A2(M8)
2007 PRINTM3$:PRINT"SQUARE";AZ
2020 GOSUB 3205:IFAF$<>" " THEN2105
2030 AA=1:IFAZ=1 THENAF$="GO"
2040 IFAZ=3 OR AZ=18 OR AZ=34 THEN AF$="COMMUNITY CHEST"
2050 IFAZ=5 THEN AF$="INCOME TAX"
2060 IFAZ=8 OR AZ=23 OR AZ=37 THEN AF$="CHANCE"
2070 IFAZ=11 THEN AF$="JAIL"
2080 IFAZ=21 THEN AF$="FREE PARKING"
2090 IFAZ=31 THEN AF$="GO TO JAIL"
2100 IFAZ=39 THEN AF$="LUXURY TAX"
2105 XF=A(AZ):IFXF>9 THEN XF=XF-20
2110 PRINT"PROPERTY -";AF$;" ";;"OWNER -"A1$(XF)
2115 IFA(AZ)>9THENPRINT"CURRENTLY MORTGAGED!"
2120 IFAZ<>11 THEN 2160
2130 AB$=" ":FORB9=1TOP:IFB(B9)>40 THEN AB$=AB$+A1$(B9)+" "
2140 NEXTB9:IFZ>40 THEN AB$=AB$+"TRS-80"
2150 PRINT"IN JAIL -";AB$:PRINT"JUST VISITING"
2160 AB$=" ":FORB9=1TOP:IFB(B9)=AZ THEN AB$=AB$+A1$(B9)+" "
2170 NEXTB9:IFZ=AZ THEN AB$=AB$+"TRS-80"
2175 GOSUB 6000:IFFZ$<>" " THEN PRINT"COLOR GROUP -";FZ$
2180 PRINT"TOKENS -";AB$
2190 IFAA=1 THEN 2355
2200 PRINT"PRICE -";AG:A9=AG/2
2210 PRINT"MORTGAGE VALUE -";A9
2220 IFAZ=13 OR AZ=29 THEN 2230 ELSE 2260
2230 PRINT"IF 1 UTILITY OWNED, RENT = 4X DICE THROWN"
2240 PRINT"IF BOTH UTILITIES OWNED, RENT = 10X DICE THROWN"
2250 GOTO 2355
2260 PRINT"RENT -";AH
2265 IFAZ=6 OR AZ=16 OR AZ=26 OR AZ=36 THEN 2270 ELSE 2300
2270 PRINT"IF 2 RR OWNED";AI;" ";
2280 PRINT"IF 3 RR OWNED";AJ;" ";
2290 PRINT"IF 4 RR OWNED";AK
2295 GOTO 2355
2300 PRINT"WITH 1 HOUSE -";AI;" ";
2310 PRINT"WITH 2 HOUSES -";AJ;" ";
2320 PRINT"WITH 3 HOUSES -";AK
2330 PRINT"WITH 4 HOUSES -";AL;" ";"WITH HOTEL -";AM;" ";
2340 IFA7(AZ)=5 THEN PRINT"HOTEL ON PROPERTY"
2345 IFA7(AZ)>0 AND A7(AZ)<5 THEN PRINT A7(AZ);" HOUSES ON PROPERTY"
2350 PRINT"IMPROVEMENT COST -";AN
2355 FORAA=1TO1000:NEXTAA
2365 RETURN
2400 REM PAY INCOME TAX
2405 MJ=200:M2=I
2410 IFA2(I)<200 THEN GOSUB 2000
2420 A2(I)=A2(I)-200
2430 IFA2(M2)<0 THEN PRINT"YOU ARE BROKE AND OUT OF THE GAME!":B(M2)=0:F=F+1:GOS
UB 7000
2440 RETURN
2600 REM PAY LUXURY TAX
2605 MJ=75:M2=I
2610 IFA2(I)<75 THEN GOSUB 2000
2620 A2(I)=A2(I)-75
```

```
2630 IFA2(I)<0THEN 2430
2640 RETURN
2800 REM MORTGAGE PROPERTIES TO GAIN CASH
2801 PRINT"PLAYER-";A1$(M2)
2805 M9$="CASH GAIN"
2807 IFM2=9THENGOSUB7100:RETURN
2810 FORM6=1TO40
2820 IFA(M6)<>M2THEN2890ELSEAZ=M6:GOSUB2000
2830 IFA7(M6)=0 THEN 2860
2840 INPUT"SELLING BUILDINGS (Y/N)";V$
2850 IFV$<>"Y" THEN 2890
2855 A7(M6)=A7(M6)-1:A2(M2)=A2(M2)+(AN/2)
2857 IFA2(I)>MJ THEN RETURN ELSE 2830
2860 INPUT"WISH MORTGAGE (Y/N)";V$
2870 IFV$<>"Y" THEN 2890
2875 A2(M2)=A2(M2)+(AG/2):A(M6)=I+100
2877 IFA2(I)>MJ THEN RETURN
2890 NEXTM6
2892 IFA2(I)<MJ THEN 2430
2894 RETURN
2990 REM SHUFFLE THE COMMUNITY CHEST DECK ROUTINE
3000 FORIZ=1TO16:A8(IZ)=0:NEXTIZ:HC=1
3005 PRINT"SHUFFLE COMMUNITY CHEST DECK"
3010 FORIZ=1TO16
3020 X=RND(16):IFX=14ANDTC=1THENX=17
3030 FORM=1TO16
3040 IFA8(M)=X THEN 3020
3050 NEXTM:A8(IZ)=X:NEXTIZ:RETURN
3090 REM SHUFFLE THE CHANCE DECK ROUTINE
3100 FORIZ=1TO16:A9(IZ)=0:NEXTIZ:HD=1
3105 PRINT"SHUFFLE CHANCE DECK"
3110 FORIZ=1TO16
3120 X=RND(16):IFX=1ANDTG=1THENX=17
3130 FORM=1TO16
3140 IFA9(M)=X THEN 3120
3150 NEXTM:A9(IZ)=X:NEXTIZ:RETURN
3200 REM ASSIGN STATISTICS OF PROPERTY (AZ) ROUTINE
3205 AF$=" ":AG=0:AH=0:AI=0:AJ=0:AK=0:AL=0:AM=0:AN=0:AA=0
3210 IFAZ=2 THEN AF$="MEDITERRANEAN AVE":AG=60:AH=2:AI=10:AJ=30:AK=90:AL=160:AM=
250:AN=50:RETURN
3215 IFAZ=1 OR AZ=3 OR AZ=5 OR AZ=18 OR AZ=34 OR AZ=8 OR AZ=23 OR AZ=37 OR AZ=11
 THEN AA=1:RETURN
3220 IFAZ=4 THEN AF$="BALTIC AVE":AG=60:AH=4:AI=20:AJ=60:AK=180:AL=320:AM=450:AN
=50:RETURN
3225 IFAZ=21ORAZ=31ORAZ=39THENAA=1:RETURN
3230 IFAZ=6 THEN AF$="READING RR":AG=200:AH=25:AI=50:AJ=100:AK=200:RETURN
3240 IFAZ=7 THENAF$="ORIENTAL AVE":AG=100:AH=6:AI=30:AJ=90:AK=270:AL=400:AM=550:
AN=50:RETURN
3250 IFAZ=9 THEN AF$="VERMONT AVE":AG=100:AH=6:AI=30:AJ=90:AK=270:AL=400:AM=550:
AN=50:RETURN
3260 IFAZ=10 THEN AF$="CONNECTICUT AVE":AG=120:AH=8:AI=40:AJ=100:AK=300:AL=450:A
M=600:AN=50:RETURN
3270 IFAZ=12THENAF$="ST. CHARLES PLACE":AG=140:AH=10:AI=50:AJ=150:AK=450:AL=625:
AM=750:AN=100:RETURN
3280 IFAZ=13THENAF$="ELECTRIC CO.":AG=150:AH=4:AI=10:RETURN
3290 IFAZ=14THENAF$="STATES AVE":AG=140:AH=10:AI=50:AJ=-150:AK=450:AL=625:AM=750
:AN=100:RETURN
3295 IFAZ=15THENAF$="VIRGINIA AVE":AG=160:AH=12:AI=60:AJ=180:AK=500:AL=700:AM=90
0:AN=100:RETURN
3300 IFAZ=16THENAF$="PENNSYLVANIA RR":AG=200:AH=25:AI=50:AJ=100:AK=200:RETURN
3310 IFAZ=17THENAF$="ST. JAMES PLACE":AG=180:AH=14:AI=70:AJ=200:AK=550:AL=750:AM
=950:AN=100:RETURN
3320 IFAZ=19THENAF$="TENNESSEE AVE":AG=180:AH=14:AI=70:AJ=200:AK=550:AL=750:AM=9
50:AN=100:RETURN
3330 IFAZ=20THENAF$="NEW YORK AVE":AG=200:AH=16:AI=80:AJ=220:AK=600:AL=800:AM=10
00:AN=100:RETURN
3340 IFAZ=22THENAF$="KENTUCKY AVE":AG=220:AH=18:AI=90:AJ=250:AK=700:AL=875:AM=10
50:AN=150:RETURN
3350 IFAZ=24THENAF$="INDIANA AVE":AG=220:AH=18:AI=90:AJ=250:AK=700:AL=875:AM=105
0:AN=150:RETURN
3360 IFAZ=25THENAF$="ILLINOIS AVE":AG=240:AH=20:AI=100:AJ=300:AK=750:AL=925:AM=1
100:AN=150:RETURN
3370 IFAZ=26THENAF$="B & O RR":AG=200:AH=25:AI=50:AJ=100:AK=200:RETURN
3380 IFAZ=27THENAF$="ATLANTIC AVE":AG=260:AH=22:AI=110:AJ=330:AK=800:AL=975:AM=1
150:AN=150:RETURN
3390 IFAZ=28THENAF$="VENTNOR AVE":AG=260:AH=22:AI=110:AJ=330:AK=800:AL=975:AM=11
50:AN=150:RETURN
3400 IFAZ=29THENAF$="WATER WORKS":AG=150:AH=4:AI=10:RETURN
3410 IFAZ=30THENAF$="MARVIN GARDENS":AG=280:AH=24:AI=120:AJ=360:AK=850:AL=1025:A
M=1200:AN=150:RETURN
3420 IFAZ=32THENAF$="PACIFIC AVE":AG=300:AH=26:AI=130:AJ=390:AK=900:AL=1100:AM=1
275:AN=200:RETURN
3430 IFAZ=33THENAF$="NO. CAROLINA AVE":AG=300:AH=26:AI=130:AJ=390:AK=900:AL=1100
:AM=1275:AN=200:RETURN
3440 IFAZ=35THENAF$="PENNSYLVANIA AVE":AG=320:AH=28:AI=150:AJ=450:AK=1000:AL=120
0:AM=1400:AN=200:RETURN
3450 IFAZ=36THENAF$="SHORT LINE RR":AG=200:AH=25:AI=50:AJ=100:AK=200:RETURN
3460 IFAZ=38THENAF$="PARK PLACE":AG=350:AH=35:AI=175:AJ=500:AK=1100:AL=1300:AM=1
500:AN=200:RETURN
3470 IFAZ=40THENAF$="BOARDWALK":AG=-400:AH=50:AI=200:AJ=600:AK=1400:AL=1700:AM=2
000:AN=200:RETURN
3480 RETURN
3600 REM COMPUTE RENT (AP) FOR PROPERTY (AZ) ROUTINE
3610 AP=0:GOSUB3205:IFA(AZ)>10THEN RETURN
3620 IFAZ<>6THEN3670
3630 AP=AH:IFA(16)=A(6)THENAP=AP*2
```

```
3640 IFA(26)=A(6)THENAP=AP*2
3650 IFA(36)=A(6)THENAP=AP*2
3660 RETURN
3670 IFAZ<>16THEN3720
3680 AP=AH:IFA(16)=A(6)THENAP=AP*2
3690 IFA(16)=A(26)THENAP=AP*2
3700 IFA(16)=A(36)THENAP=AP*2
3710 RETURN
3720 IFAZ<>26THEN3770
3730 AP=AH:IFA(26)=A(6)THENAP=AP*2
3740 IFA(26)=A(16)THENAP=AP*2
3750 IFA(26)=A(36)THENAP=AP*2
3760 RETURN
3770 IFAZ<>36THEN3820
3780 AP=AH:IFA(36)=A(6)THENAP=AP*2
3790 IFA(36)=A(16)THENAP=AP*2
3800 IFA(36)=A(26)THENAP=AP*2
3810 RETURN
3820 IFAZ=13ORAZ=29THEN3830ELSE3880
3830 AP=AH:IFA(13)=A(29)THENAP=AI
3840 X=INT(RND(0)*6):IFX=0THENX=1
3850 Y=INT(RND(0)*6):IFY=0THENY=1
3860 PRINT"THE DICE ROLL FOR RENT WAS ";X;Y
3870 AP=AP*(X+Y):RETURN
3880 AP=AH:IFA7(AZ)=1THENAP=AI
3890 IFA7(AZ)=2THENAP=AJ
3900 IFA7(AZ)=3THENAP=AK
3910 IFA7(AZ)=4THENAP=AL
3920 IFA7(AZ)=5THENAP=AM
3930 RETURN
4000 REM DRAW A COMMUNITY CHEST CARD ROUTINE
4010 HC=HC+1:IFHC>16THENGOSUB3000
4020 AW=A8(HC):BG=0:BH=0:BI=0:BJ=0:BW=1
4030 IFAW=1THENBG=100:PRINT"XMAS FUND MATURES, COLLECT $100":RETURN
4040 IFAW=2THENBI=1:PRINT"GO DIRECTLY TO JAIL!":RETURN
4050 IFAW=3THENBG=-150:PRINT"PAY SCHOOL TAX OF $150":RETURN
4060 IFAW=4THENBG=25:PRINT"RECEIVE FOR SERVICES $25":RETURN
4070 IFAW=5THENBG=50:BJ=1:PRINT"GRAND OPERA OPENING,COLLECT $50 FROM EVERY PLAYE
R":RETURN
4080 IFAW=6THENBG=10:PRINT"YOU HAVE WON SECOND PRIZE IN A BEAUTY CONTEST, COLLEC
T $10":RETURN
4090 IFAW=7THENBG=20:PRINT"INCOME TAX REFUND, COLLECT $20":RETURN
4100 IFAW=8THENBG=100:PRINT"LIFE INSURANCE MATURES, COLLECT $100":RETURN
4110 IFAW=9THENBG=45:PRINT"FROM SALE OF STOCK YOU GET $45":RETURN
4120 IFAW=10THENBG=100:PRINT"YOU INHERIT $100":RETURN
4130 IFAW<>11THEN4180
[illegible] PRINT"YOU ARE ASSESSED FOR STREET REPAIRS, $40 PER HOUSE, $115 PER HOTEL"
4140 FORBM=1TO40:IFA(BM)<>ITHEN4170
4150 IFA7(BM)=5THENBG=BG+115:GOTO4170
4160 IFA7(BM)>0THENBG=BG+(A7(BM)*40)
4170 NEXTBM:BG=BG*-1:RETURN
4180 IFAW=12THENBH=1:PRINT"ADVANCE TO GO":RETURN
4190 IFAW=13THENBG=-100:PRINT"PAY HOSPITAL $100":RETURN
4200 IFAW=14THENBI=2:PRINT"GET OUT OF JAIL FREE CARD":RETURN
4210 IFAW=15THENBG=200:PRINT"BANK ERROR IN YOUR FAVOR, COLLECT $200":RETURN
4220 IFAW=16THENBG=-50:PRINT"DOCTOR'S FEE, PAY $50":RETURN
4230 IFAW=17THEN4010
4240 RETURN
4300 REM DRAW A CHANCE CARD ROUTINE
4310 HD=HD+1:IFHD>16THENGOSUB3100
4320 AW=A9(HD):BG=0:BH=0:BI=0:BJ=0:BW=1
4330 IFAW=1THENBI=3:PRINT"GET OUT OF JAIL FREE CARD":RETURN
4335 IFAW=11THEN4360
4340 IFAW=17THEN4310
4350 IFAW<>2THEN4430
4360 PRINT"ADVANCE TO THE NEAREST RAILROAD AND PAY OWNER TWICE THE RENTAL HE IS
ENTITLED":BW=2
4370 IFM=8THENBH=16
4380 IFM=23THENBH=26
4390 IFM=37THENBH=6
4420 RETURN
4430 IFAW=3THENPRINT"ADVANCE TO GO":BH=1:RETURN
4440 IFAW=4THENBG=50:BJ=2:PRINT"YOU HAVE BEEN ELECTED CHAIRMAN OF THE BOARD, PAY
 EACH PLAYER $50":RETURN
4450 IFAW=5THENBG=-15:PRINT"PAY POOR TAX OF $15":RETURN
4460 IFAW=6THENBH=M-3:PRINT"GO BACK 3 SPACES"
4470 IFAW=6ANDBH<1THENBH=BH+40
4480 IFAW=6THENRETURN
4490 IFAW=7THENBH=12:PRINT"ADVANCE TO ST. CHARLES PLACE"
4510 IFAW=7THENRETURN
4520 IFAW<>8THEN4560
4525 PRINT"MAKE GENERAL REPAIRS ON ALL YOUR PROPERTY, FOR EACH HOUSE PAY $25, FO
R EACH HOTEL PAY $100"
4530 FORBM=1TO40:IFA(BM)<>ITHEN4550
4540 IFA7(BM)=5THENBG=BG+100ELSEBG=BG+(A7(BM)*25)
4550 NEXTBM:BG=BG*-1:RETURN
4560 IFAW=9THENBH=40:PRINT"TAKE A WALK ON THE BOARDWALK":RETURN
4570 IFAW=10THENBG=50:PRINT"BANK PAYS YOU DIVIDEND OF $50":RETURN
4580 IFAW<>12THEN4610
4585 PRINT"TAKE A RIDE ON THE READING RAILROAD"
4590 BH=6:RETURN
4610 IFAW=13THENBI=1:PRINT"GO DIRECTLY TO JAIL":RETURN
4620 IFAW=14THENBG=150:PRINT"YOUR BUILDING AND LOAN MATURES, COLLECT $150":RETUR
N
4630 IFAW=15THENBH=25:PRINT"ADVANCE TO ILLINOIS AVE"
4650 IFAW=15THENRETURN
```

```
4660 IFAW<>16THENRETURN
4665 PRINT"ADVANCE TO NEAREST UTILITY AND PAY 10 TIMES THE THROW OF THE DICE"
4670 IFM=8ORM=37THENBH=13
4680 IFM=23THENBH=29
4685 BW=10
4740 RETURN
5000 REM PAY RENT TO A PLAYER
5010 M6=A(M):PRINT"YOU OWE RENT TO ";A1$(M6)
5020 AZ=M:GOSUB3600:M2=I:MJ=AP*BW
5022 PRINT"RENT ";MJ:GOSUB2355
5030 IFMJ>A2(I)THENGOSUB2000
5032 IFB(I)=0THEN5070
5040 A2(I)=A2(I)-(AP*BW)
5050 IFA2(I)<0THEN2430
5060 A2(M6)=A2(M6)+(AP*BW)
5070 RETURN
6000 REM DETERMINE COLOR GROUP OF PROPERTY (AZ) IN (FZ$)
6005 FZ$=" ":FY=0
6010 IFAZ=2ORAZ=4THEN6020ELSE6050
6020 FZ$="PURPLE"
6030 IFA(2)=A(4)THENFY=A(2)
6040 RETURN
6050 IFAZ=7ORAZ=9ORAZ=10THEN6060ELSE6090
6060 FZ$="LT BLUE"
6070 IFA(7)=A(9)ANDA(9)=A(10)THENFY=A(7)
6080 RETURN
6090 IFAZ=12ORAZ=14ORAZ=15THEN6100ELSE6130
6100 FZ$="VIOLET"
6110 IFA(12)=A(14)ANDA(14)=A(15)THENFY=A(12)
6120 RETURN
6130 IFAZ-17ORAZ=19ORAZ=20THEN6140ELSE6170
6140 FZ$="ORANGE"
6150 IFA(17)=A(19)ANDA(19)=A(20)THENFY=A(17)
6160 RETURN
6170 IFAZ=22ORAZ=24ORAZ=25THEN6180ELSE6210
6180 FZ$="RED"
6190 IFA(22)=A(24)ANDA(24)=A(25)THENFY=A(22)
6200 RETURN
6210 IFAZ=27ORAZ=28ORAZ=30THEN6220ELSE6250
6220 FZ$="YELLOW"
6230 IFA(27)=A(28)ANDA(28)=A(30)THENFY=A(27)
6240 RETURN
6250 IFAZ=32ORAZ=33ORAZ=35THEN6260ELSE6290
6260 FZ$="GREEN"
6270 IFA(32)=A(33)ANDA(33)=A(35)THENFY=A(32)
6280 RETURN
6290 IFAZ=38ORAZ=40THEN6300ELSE6330
6300 FZ$="BLUE"
```

```
6310 IFA(38)=A(40)THENFY=A(38)
6320 RETURN
6330 IFAZ=6ORAZ=16ORAZ=26ORAZ=36THENFZ$="RAILROADS"
6340 IFAZ=13ORAZ=29THENFZ$="UTILITIES"
6350 RETURN
6500 REM ALLOW PLAYER TO UNMORTGAGE PROPERTIES, BUY HOUSES AND HOTELS ROUTINE
6501 IFI=9THENGOSUB7200:RETURN
6510 FORM=1TO40
6515 IFA(M)=ITHEN6520ELSE6530
6520 AZ=M:GOSUB3205:GOSUB6000
6525 IFFY=IANDA2(I)>=ANANDA7(M)<5THEN6540
6530 IFA(M)=I+100ANDA2(I)>AG/2THEN6540
6535 NEXTM:RETURN
6540 INPUT"ENTER 1 TO UNMORTGAGE, 2 TO IMPROVE PROPERTIES OR 3 TO END YOUR TURN"
;M:IFM<1ORM>3THEN6540
6545 IFM=3THENRETURNELSEK=M
6550 INPUT"PROPERTY SQUARE NO. ";M:IFM<1ORM>40THEN6550
6552 IFK=2THEN6585
6555 IFA(M)=I+100THEN6565
6560 PRINT"NOT MORTGAGED!":GOTO6540
6565 AZ=M:GOSUB3205
6570 IFA2(I)>=AG/2THEN6580
6575 PRINT"INSUFFICIENT FUNDS!":GOTO6540
6580 A2(I)=A2(I)-(AG/2):A(M)=I:GOTO6540
6585 IFA(M)=ITHEN6595
6590 PRINT"NOT OWNED!":GOTO6540
6595 IFA7(M)<5THEN6605
6600 PRINT"HOTEL ESTABLISHED!":GOTO6540
6605 AZ=M:GOSUB3205:GOSUB6000
6610 IFFY=ITHEN6620
6615 PRINT"COLOR GROUP NOT OWNED!":GOTO6540
6620 IFA2(I)>=ANTHEN6625ELSE6575
6625 A2(I)=A2(I)-AN:A7(M)=A7(M)+1
6630 IFA7(M)=5THEN PRINT"HOTEL"ELSEPRINTA7(M);"HOUSES"
6635 GOTO6540
7000 FORM4=1TO40:IFA(M4)=IORA(M4)=I+100THENA(M4)=0:A7(M4)=0
7010 NEXTM4:RETURN
7100 REM MORTGAGE PROPERTIES FOR COMPUTER
7140 FORM3=1TO5:M7=6-M3:FORM4=1TO40
7149 IFA(M4)=M2ANDA7(M4)=M7THENPRINT"UNIMPROVED PROPERTY ";M4
7150 IFA(M4)=M2ANDA7(M4)=M7THENA7(M4)=A7(M4)-1:AZ=M4:GOSUB3205:A2(M2)=A2(M2)+(AN
/2)
7155 IFA2(M2)>MJTHENRETURN
7160 NEXTM4:NEXTM3
7170 FORM4=1TO40
7174 IFA(M4)=M2ANDA7(M4)=0THENPRINT"MORTGAGED PROPERTY ";M4
7175 IFA(M4)=M2ANDA7(M4)=0THENAZ=M4:GOSUB3205:A2(M2)=A2(M2)+(AG/2):A(M4)=I+100
7180 IFA2(M2)>MJTHENRETURN
```

```
7185 NEXTM4
7190 IFA2(M2)<MJTHEN2430
7195 RETURN
7200 REM UNMORTGAGE AND IMPROVE PROPERTIES FOR THE COMPUTER
7205 Z0=0
7207 FORM4=1TO40
7208 IFA(M4)<>9ANDA(M4)<>109THEN7230
7210 A2=M4:GOSUB3205:GOSUB6000
7212 IFFY=9THENZ0=Z0+1
7215 IFA(M4)<>I+100THEN7230
7219 IFA2(I)>AG/2THENPRINT"UNMORTGAGED PROPERTY ";M4
7220 IFA2(I)>AG/2THENA(M4)=I:A2(I)=A2(I)-(AG/2)
7230 NEXTM4
7235 IFZ0=0THENRETURN
7240 FORM3=1TO5:FORM4=1TO40:A2=-M4:GOSUB3205:GOSUB6000
7250 IFFY<>1ORA7(M4)<>M3-1THEN7260
7254 IFA2(I)>ANTHENPRINT"IMPROVED PROPERTY ";M4;" TO ";M3
7255 IFA2(I)>ANTHENA2(I)=A2(I)-AN:A7(M4)=A7(M4)+1
7260 NEXTM4:NEXTM3:RETURN
```

A WORD SEARCH PUZZLE PROGRAM - WITH A TWIST!

Here is a WORD SEARCH PUZZLE program that goes beyond the scope and capabilities of most word search puzzle generators currently available on the market. To begin with, this program not only generates word search puzzles but can be given a word search puzzle from an external source and the program will solve the puzzle for you! Within its capability to generate a word search puzzle, many options are available.

When the program starts, it asks you what screen mode you would like. In the normal screen mode, only 60 characters across and 12 rows down can be displayed at one time. In the 32 character mode, only 28 characters across and 10 rows down can be displayed at one time. In both modes, neither the across nor the down screen display restrictions in any way affect the size of the puzzle you can build. Also, both modes provide a scale for both across and down display.

Next, the program asks what type of line printer might be used. The reason for this is that this program can be set up to both display and print your puzzle so that you can optionally work on the puzzle without the computer at a later time. If no printer is to be used, you should choose the 80 character per line option since that is the maximum number of characters across allowed in any puzzle. The other printer options, 20 characters per line and 40 characters per line, are for the larger pitch settings on a Quick Printer attached to your TRS-80. The option selected here is used to restrict the maximum number of characters allowed across in your puzzle.

Now you must enter the actual size of your puzzle. The across dimension is restricted by the printer option. The down dimension is restricted to 80 rows maximum. While this program is designed to run on a Level 2 TRS-80 with 16K, a puzzle larger than 50 by 50 will require 32K.

Following this, the computer will want to know if you are giving it a puzzle of the size specified to solve. Entering Y will indicate yes. If yes, you will be asked to enter your puzzle a row at a time starting at row 1. Each row entered must be the correct number of characters across. If the length is incorrect, an error message is displayed and you must re-enter that row.

After all rows in the puzzle are entered, the computer will ask what words you wish to find. If the word is found, the starting row and column are displayed along with the direction the word is layed out in the puzzle. These directions may be: across, down, left diagonal, right diagonal, backwards across, backwards up, left backwards diagonal, right backwards diagonal. These eight directions are also used in generating a puzzle. If the word cannot be found, the program will tell you so and ask for another word. When you have asked for all the words you wish, enter an * to end the program.

If generating a puzzle, the program will ask if the puzzle should be printed. Enter a Y to indicate yes. Next, the program will ask for all the words you wish included in the generated puzzle. The length of each word must conform to maximum characters specified in the printer option specified earlier (20, 40 or 80 characters long). If the puzzle is to be printed, each word selected is listed on the printer. If the length of the word is incorrect, an error message is displayed and another word requested. When all your words are inputted, enter an * to have the puzzle generated. During this time, if the puzzle is to be printed, the puzzle will be printed a row at a time.

Once the puzzle is generated, the program asks what portion of the puzzle you would like displayed on the screen. Taking the screen mode and puzzle size options into account, the program uses the column and row of the letter to appear in the upper left hand display position on the screen. The program will automatically truncate the display properly so that you may set the upper left hand corner column and row to any position in the puzzle.

Once the display is complete, you may enter an E to end the word search portion of the program, L to redisplay a new section of the puzzle or G to guess a word's position and direction in the puzzle. In guessing, you are asked for the column and row of the first letter of a word in the puzzle, the word and which direction the word is layed out in the puzzle. You are told if your guess is correct or not. When you enter E, the program will tell you how many mistakes you made and how many hidden words you have not found yet. Be careful in guessing a word's position! Random letter inserts in the puzzle generation by the program may accidentally form a word you asked to have hidden. Only the position of originally hidden words are valid in a guess!

If all the words hidden are not found, the program will automatically engage the puzzle solving portion of the program to help you find words in the generated puzzle.

With all these features, we hope you enjoy this unusual version of WORD SEARCH PUZZLE programming presented in this article. To save inputting this program, you may obtain a CLOAD tested cassette of this program only from THE 80 NOTEBOOK for the modest price of only $4.95.

WORD SEARCH PUZZLE PROGRAM LISTING:

```
5 CLEAR 300
10 REM WORD SEARCH PUZZLES
20 REM COPYRIGHT 1980, THE 80 NOTEBOOK
30 DEFINTA-Z:RANDOM:CLS:PRINT"ENTER PUZZLE SIZE"
32 INPUT"ENTER 1=NORMAL SCREEN,2=32 CHARACTER MODE";N9
33 IFN9<1ORN9>2THEN32
34 N3=60:IFN9=2THENN3=28
35 INPUT"ENTER 1=20 CPL PRINTER,2=40 CPL PRINTER,3=80 CPL PRINTER";N7
36 IFN7<1ORN7>3THEN35
37 N5$=CHR$(32-N7):N6=80:IFN7=1THENN6=20
38 IFN7=2THENN6=40
40 INPUT"HOW MANY LETTERS ACROSS";X:IFX<1ORX>80 THEN40
45 IFX>N6THEN40
50 INPUT"HOW MANY LETTERS DOWN";Y:IFY<1ORY>80 THEN 50
60 DIMA(Y,X):P=15360
65 U9=0
70 FORI=0TOY:FORW=0TOX:A(I,W)=0:NEXTW:NEXTI
90 INPUT"ARE YOU GIVING ME A PUZZLE TO SOLVE (Y/N)";B$
100 IFB$="Y"THEN 2000
110 INPUT"SHOULD I PRINT THE PUZZLE ON THE LINE PRINTER";C$
120 IFC$="Y"THENLPRINTN5$;"WORD SEARCH PUZZLE":LPRINTN5$;" ":LPRINTN5$;"THE WORD
S TO FIND-":LPRINTN5$;" "
130 PRINT"PLEASE ENTER THE WORDS YOU WISH TO INCLUDE IN THE PUZZLE"
140 PRINT"ENTER * WHEN DONE"
150 INPUTF$:IFF$="*"THEN 600
155 IFC$="Y"THENLPRINTF$
160 L=LEN(F$):IFL<1 THEN 150
161 IFL>N6THEN150
162 IFL>XANDL>YTHENPRINT"TOO LONG!":GOTO150
165 U9=U9+1
170 R=RND(X):T=RND(Y):IFA(T,R)<>0 THEN 170
190 J=RND(8):ONJGOTO200,300,400,500,3100,3200,3300,3400
200 IFL>X-R+1 THEN 170
210 FORI=0TOL-1:IFA(T,R+I)=0 THEN 230
```

```
220 M$=MID$(F$,I+1,1):IFA(T,R+I)<>ASC(M$)+100 THEN 170
230 NEXTI
240 FORI=1TOL:M$=MID$(F$,I,1):N=ASC(M$)+100
250 A(T,R+I-1)=N:NEXTI:GOTO150
300 IFL>Y-T+1 THEN 170
305 IFL>X-R+1 THEN 170
310 FORI=0TOL-1
330 IFA(T+I,R+I)=0 THEN 350
340 M$=MID$(F$,I+1,1):IFA(T+I,R+I)=ASC(M$)+100 THEN170
350 NEXTI
360 FORI=0TOL-1:Q=(I*(X+1))+K
370 M$=MID$(F$,I+1,1):A(T+I,R+I)=ASC(M$)+100:NEXTI:GOTO150
400 IFL>Y-T+1 THEN 170
410 FORI=0TOL-1:Q=(I*X)+K
420 IFA(T+I,R)=0 THEN 440
430 M$=MID$(F$,I+1,1):IFA(T+I,R)<>ASC(M$)+100 THEN 170
440 NEXTI
450 FORI=0TOL-1:Q=(I*X)+K
460 M$=MID$(F$,I+1,1):A(T+I,R)=ASC(M$)+100:NEXTI:GOTO 150
500 IFL>Y-T+1 THEN 170
505 IFR<L THEN 170
510 FORI=0TOL-1:Q=(I*(X-1))+K
520 IFA(T+I,R-I)=0 THEN 540
530 M$=MID$(F$,I+1,1):IFA(T+I,R-I)<>ASC(M$)+100 THEN 170
540 NEXTI
550 FORI=0TOL-1:Q=(I*(X-1))+K
560 M$=MID$(F$,I+1,1):A(T+I,R-I)=ASC(M$)+100:NEXTI:GOTO150
600 FORI=0TOY:FORW=0TOX:IFA(I,W)=0 THEN A(I,W)=RND(26)+64
610 NEXTW
620 NEXTI:U=0
630 IFC$<>"Y" THEN 700
635 LPRINTN5$;" ":LPRINTN6$;"YOUR PUZZLE-":LPRINTN5$;" "
636 R=P
640 FORI=1TOY:F$="":FORT=1TOX:N=PEEK(R)
642 N=A(I,T):R=P
645 IFN>100 THEN N=N-100
650 R=R+1:F$=F$+CHR$(N):NEXTT
660 LPRINTN5$;F$:NEXTI:LPRINTCHR$(29);" "
700 CLS:INPUT"ENTER UPPER LEFT CORNER COLUMN, ROW";R,T
710 IFR<1ORR>XORT<1ORT>Y THEN 700 ELSE CLS
720 IFN9=2THENPRINTCHR$(23);
725 PRINTSTRING$(4," ");
730 N=R+N3-1:IFN>XTHENN=X
740 FORI=RTON:J=INT(I/10)+48:PRINTCHR$(J);:NEXTI
750 IFN-R+1<N3THENPRINT" "
760 PRINTSTRING$(4," ");:FORI=RTON
765 J=I
770 J=J-(INT(J/10)*10)+48:PRINTCHR$(J);:NEXTI
780 IFN-R+1<N3THENPRINT" "
790 W=((T-1)*X)+R:G=T+11:IFG>Y THEN G=Y
792 IFN9=1THEN800
794 G=T+9:IFG>YTHENG=Y
800 FORI=TTOG:Z=W
810 F$=STR$(I)+" ":IFLEN(F$)<4 THENF$=F$+" "
815 Z=0:W=0
820 PRINTF$;:FORK=RTON:V=PEEK(P+W):W=W+1
822 V=A(I,K)
825 IFV>100THENV=V-100
830 PRINTCHR$(V);:NEXTK:IFN-R+1<N3THENPRINT" "
840 Z=Z+X:W=Z:NEXTI
850 PRINT"ENTER L=LIST PUZZLE SECTION, G=GUESS WORD POSITION, E=END"
860 F$="":F$=INKEY$:IFF$=""ORF$=" " THEN 860
865 IFF$="E" THEN 1600
870 IFF$="L" THEN 700
880 IFF$<>"G" THEN 860
890 CLS:INPUT"ENTER COLUMN, ROW OF FIRST LETTER IN WORD";R,T
900 IFR<1 OR R>X OR T<1 OR T>Y THEN 890
910 INPUT"WORD";F$:L=LEN(F$):K=((T-1)*X)+R+P
915 PRINT"ENTER 5=BACKWARDS ACROSS,6=BACKWARDS LEFT DIAGONAL,7=UP BACKWARDS,8=BA
CKWARDS RIGHT DIAGONAL"
920 INPUT"1=ACROSS, 2=RIGHT DIAGONAL, 3=DOWN, 4=LEFT DIAGONAL";Q
930 IFQ<1ORQ>8THEN920
935 K=P
940 ONQGOTO1000,1100,1200,1300,3500,3600,3700,3800
950 GOTO 920
1000 IFL>X-R+1 THEN 890
1005 FORI=0TOL-1:N=PEEK(K):K=K+1:M$=MID$(F$,I+1,1)
1006 N=A(T,R+I):K=P
1007 Q=ASC(M$)+100
1010 IFQ<>NTHEN1400
1015 NEXTI:FORI=0TOL-1:A(T,R+I)=A(T,R+I)-100:NEXTI:GOTO1500
1100 IFL>Y-T+1 THEN 890
1102 IFL>X-R+1 THEN 890
1110 FORI=1TOL:N=PEEK(K):K=K+X+1:M$=MID$(F$,I,1)
1112 N=A(T+I-1,R+I-1):K=P
1120 Q=ASC(M$)+100
1130 IFQ<>N THEN 1400
1140 NEXTI:FORI=0TOL-1:A(T+I,R+I)=A(T+I,R+I)-100:NEXTI:GOTO1500
1200 IFL>Y-T+1 THEN 890
1210 FORI=1TOL:N=PEEK(K):K=K+X:M$=MID$(F$,I,1)
1212 N=A(T+I-1,R):K=P
1220 Q=ASC(M$)+100
1230 IFQ<>N THEN 1400
1240 NEXTI:FORI=0TOL-1:A(T+I,R)=A(T+I,R)-100:NEXTI:GOTO1500
```

```
1300 IFL>Y-T+1 THEN 890
1302 IFR<L THEN 890
1310 FORI=1TOL:N=PEEK(K):K=K+X-1:M$=MID$(F$,I,1)
1312 N=A(T+I-1,R-I+1):K=P
1320 O=ASC(M$)+100
1330 IFO<>N THEN 1400
1340 NEXTI:FORI=0TOL-1:A(T+I,R-I)=A(T+I,R-I)-100:NEXTI:GOTO1500
1400 U=U+1:PRINT"NOT A HIDDEN WORD!"
1410 FORI=1TO1000:NEXTI
1420 GOTO 850
1500 PRINT"CORRECT"
1505 U9=U9-1
1510 GOTO 1410
1600 PRINT"YOU HAD";U;"ERRORS!"
1605 IFU9>0THENPRINT"YOU STILL MUST FIND";U9;" WORDS!":GOSUB3900:GOTO2032
1610 END
2000 PRINT"ENTER YOUR PUZZLE A ROW AT A TIME":K=P
2005 FORI=1TOY
2010 INPUTF$:IFLEN(F$)<>X THEN PRINT"INVALID LENGTH":GOTO 2010
2020 FORJ=1TOX:K=K+1:M$=MID$(F$,J,1):N=ASC(M$)
2030 A(I,J)=N:K=P:NEXTJ:NEXTI
2032 PRINT"ENTER * TO END THE SESSION"
2040 INPUT"WHAT WORD SHALL I LOOK FOR";F$:L=LEN(F$)
2050 IFF$="*" THEN PRINT"THANK YOU!":END
2060 M$=MID$(F$,1,1):N=ASC(M$):K=P
2070 FORI=1TOY
2080 FORJ=1TOX
2090 IFN<>A(I,J) THEN 2240
2092 K=P
2095 G=K
2097 U8=0:GOSUB4000:IFU8=1THEN2040
2100 IFL>X-J+1 THEN 2140
2110 FORR=0TOL-1:C$=MID$(F$,R+1,1):T=ASC(C$):Z=PEEK(K+R)
2112 Z=A(I,J+R)
2120 IFT<>Z THEN 2140
2130 NEXTR:PRINT"ROW";I;"COLUMN";J;"ACROSS":GOTO 2040
2140 IFL>Y-I+1 THEN 2240
2142 IFL>X-J+1 THEN 2180
2150 FORR=0TOL-1:C$=MID$(F$,R+1,1):T=ASC(C$):Z=PEEK(G)
2155 Z=A(I+R,J+R):G=P
2160 G=G+X+1:IFT<>Z THEN 2180
2170 NEXTR:PRINT"ROW";I;"COLUMN";J;"RIGHT DIAGONAL":GOTO 2040
2180 FORR=0TOL-1:C$=MID$(F$,R+1,1):T=ASC(C$):Z=PEEK(G)
2185 Z=A(I+R,J):G=P
2190 G=G+X:IFT<>Z THEN 2210
2200 NEXTR:PRINT"ROW";I;"COLUMN";J;"DOWN":GOTO 2040
2210 FORR=0TOL-1:C$=MID$(F$,R+1,1):T=ASC(C$):Z=PEEK(G)
2212 IFJ<L THEN 2240
2213 Z=A(I+R,J-R):G=P
2220 G=G+X-1:IFT<>Z THEN 2240
2230 NEXTR:PRINT"ROW";I;"COLUMN";J;"LEFT DIAGONAL":GOTO 2040
2240 NEXTJ:NEXTI:PRINT"I CAN NOT FIND THAT ONE!":GOTO 2040
3100 IFR<LTHEN170
3110 FORI=0TOL-1:IFA(T,R-I)=0THEN3140
3120  M$=MID$(F$,I+1,1):O=ASC(M$)+100
3130 IFA(T,R-I)<>0THEN170
3140 NEXTI
3150 FORI=0TOL-1:M$=MID$(F$,I+1,1):A(T,R-I)=ASC(M$)+100
3160 NEXTI:GOTO150
3200 IFR<LTHEN170
3210  IFT<LTHEN170
3220 FORI=0TOL-1:IFA(T-I,R-I)=0THEN3240
3230 M$=MID$(F$,I+1,1):IFA(T-I,R-I)<>ASC(M$)+100THEN170
3240 NEXTI
3250 FORI=0TOL-1:M$=MID$(F$,I+1,1):A(T-I,R-I)=ASC(M$)+100
3260 NEXTI:GOTO150
3300 IFT<LTHEN170
3310 FORI=0TOL-1:IFA(T-I,R)=0THEN3330
3320  M$=MID$(F$,I+1,1):IFA(T-I,R)<>ASC(M$)+100THEN170
3330  NEXTI
3340  FORI=0TOL-1:M$=MID$(F$,I+1,1):A(T-I,R)=ASC(M$)+100
3350 NEXTI:GOTO150
3400 IFT<LTHEN170
3410  IFL>X-R+1THEN170
3420 FORI=0TOL-1:IFA(T-I,R+I)=0THEN3440
3430 M$=MID$(F$,I+1,1):IFA(T-I,R+I)<>ASC(M$)+100THEN170
3440 NEXTI
3450 FORI=0TOL-1:M$=MID$(F$,I+1,1):A(T-I,R+I)=ASC(M$)+100
3460 NEXTI:GOTO150
3500 IFR<LTHEN890
3510 FORI=0TOL-1:M$=MID$(F$,I+1,1):IFA(T,R-I)<>ASC(M$)+100THEN1400
3520 NEXTI:FORI=0TOL-1:A(T,R-I)=A(T,R-I)-100:NEXTI:GOTO1500
3600 IFR<LTHEN890
3610 IFT<LTHEN890
3620 FORI=0TOL-1:M$=MID$(F$,I+1,1):IFA(T-I,R-I)<>ASC(M$)+100THEN1400
3630  NEXTI:FORI=0TOL-1:A(T-I,R-I)=A(T-I,R-I)-100:NEXTI:GOTO1500
3700 IFT<LTHEN890
3710 FORI=0TOL-1:M$=MID$(F$,I+1,1):IFA(T-I,R)<>ASC(M$)+100THEN1400
3720 NEXTI:FORI=0TOL-1:A(T-I,R)=A(T-I,R)-100:NEXTI:GOTO1500
3800 IFT<LTHEN890
3810 IFL>X-R+1THEN890
3820 FORI=0TOL-1:M$=MID$(F$,I+1,1):IFA(T-I,R+I)<>ASC(M$)+100THEN1400
3830 NEXTI:FORII=0TOL-1:A(T-I,R+I)=A(T-I,R+I)-100:NEXTI:GOTO1500
3900 FORI=0TOY:FORW=0TOX:IFA(I,W)>100THENA(I,W)=A(I,W)-100
```

```
3910 NEXTW:NEXTI:RETURN
4000  IFJ<LTHEN4030
4010 FORR=0TOL-1:M$=MID$(F$,R+1,1):IFA(I,J-R)<>ASC(M$)THEN4030
4020 NEXTR:PRINT"ROW";I;" COLUMN";J;" BACKWARDS ACROSS":UB=1:GOTO4140
4030 IFJ<LTHEN4070
4040 IFI<LTHEN4070
4050 FORR=0TOL-1:M$=MID$(F$,R+1,1):IFA(I-R,J-R)<>ASC(M$)THEN4070
4060 NEXTR:PRINT"ROW";I;" COLUMN";J;" BACKWARDS LEFT DIAGONAL":UB=1:GOTO4140
4070 IFI<LTHEN4100
4080 FORR=0TOL-1:M$=MID$(F$,R+1,1):IFA(I-R,J)<>ASC(M$)THEN4100
4090 NEXTR:PRINT"ROW";I;" COLUMN";J;" UP BACKWARDS":UB=1:GOTO4140
4100 IFI<LTHEN4140
4110 IFL>X-J+1THEN4140
4120 FORR=0TOL-1:M$=MID$(F$,R+1,1):IFA(I-R,J+R)<>ASC(M$)THEN4140
4130 NEXTR:PRINT"ROW";I;" COLUMN";J;" BACKWARDS RIGHT DIAGONAL":UB=1
4140 RETURN
```

* * * * * * * * * * * * * * * * *

```
WORD SEARCH PUZZLE

THE WORDS TO FIND-

ASSEMBLY
BASIC
BYTE
CASSETTE
COMPUTER
DISKETTE
DISPLAY
EXPANSION
HARDWARE
INTERFACE
KEYBOARD
KEYPAD
LANGUAGE
LISTING
MICROCOMPUTER
MONITOR
PLAYER
PRINTER
PROCESSOR
PROGRAM
QUICK
RADIO
RECORDER
SCREEN
SHACK
SOFTWARE
SOURCE
VIDEO
```

```
HHGAIECIESKFWGAVQFHEDJBGTXGMLANGUAGEMWAA
WFWYQUXAFTDEFAZMCJVOJEIQYVDIIIRFWMQOFEWH
MLUXXNTLTIRQYTHNCBFQUNUACKHJVDGJUYVVZQMO
REQHRJWLSZZOMPEZOWXRORLUYKCAHSBBDYCDEVUT
CXORSIIOWHEHREADRBNEJPJFXMFSLAFXEVITPGOR
RYFESAPFGFIGRDSDCMSMSSXZPRHPUOWRCHFQFGLT
BNSHITNXFRNCJMJIQDFICTRTASZEHQQTABDIRZAP
YPHVCMENDBSRSVEEVODOIKRWJSWNJWHRJZVGAKDV
TMODJDZJWCTPSTFJYKDXHOEFWURXFKOLSWWHTTCD
EDDFFWCREYALPODLIPEWVMHKZXHSIWXPEZKIQRKM
XXSOLREVJTVCSGZGNITSILJZEFETAYJTHXJSENZZ
HZNLFWRUGQMARGORPGPYGYYXGOBRHQKNFRAPVQVB
HCZUUIANEHHKODCZYNWCBYPGIOEXNPMIOBCAZXJL
BJLYUHWWLXJKRPZAOXUZVARWWNFDOVTZVMBAFPUU
ONEZSXTKONZACRQETVMINFPPGGTMONITORBXWDGU
BOSLBYFXVAOAKIGFHDESTNOWMIIERCGJXPUXXQYD
UTYQFJOUABIBANHHVYIJHYXOKTTMRKJEGIAZWASF
LXNYSGSZYBWXWTRCMOLPXXWRPLGKBFPKRTDJVXDQ
OADYRXHEZRYFKEYENEJBIWMGTZQMBKAEGBLSNVQH
KTHWAKKEKIPBFRESTUHIMOCXQZEMKKHCLZUZQOTZ
MDZWINGVSFQDBJRAXUQIWEUJOTNDGKLHEQXWYPLJ
MMKKESABCDHJYIGZFVPYTKSGYZZBQYFZLIMNRMRW
DCQGJEDQSUZZTTPCCKUMTFFSCINMZIBLJJHHJWIR
SDJPOITFBCTVDTEHDVKHOOFRAGZLKPROCESSORCD
QDYKREITINIJUWFCGZBCUCOCOUYKUNCHPYNBYWZR
OQTCETJUEJGSIVYRRWYYYMOSOEHPJVNTJVORETDF
MQRODTYYSKGRAJGAGUGQUJWRGPDCXDZTLCEMYTSR
HBZMRERXYFSQABIBFJOIEXPPCQYIVSOKWHTDGQDI
KVFPOSMUYWDISHMVMQUSLMAPHIHHVXDWYWIJMNOD
LBGUCSDGDVYXDQTPGPCBDJHZIYMCTAVTFFSRHVXI
HSNTEAJXXHVZLLKCIUQSEAXMEGRYTYOFMSZJNYHD
QRCERCXKGZVARJOSQURGLEYSSRKPOKXBTVRMSARA
YJLRALNUNXQNMLNIOHSCWDBLUGCGFGIXMWAJJFAR
JUZTHQVGAPJZPYCGNISODTDWTBNJJLWLEIMBYOEZ
```

BEAT THE DEVIL

by John R. Warrick TRS-80 Level II

BEAT THE DEVIL was an easy game to develop. The basic idea came from the TV game show, TIC-TAC-DOUGH where contestants try to beat the Dragon and win prizes.

The game board is a simple Tic-Tac-Toe design of 9 squares. Each square is blank until its number is selected by the player. Then either a score or a Devil graphic will show up. Although in the program itself, there are 9 values for the 9 squares, the Devil will replace one of the values during play.

Once the player is asked to CHOOSE A SQUARE the Devil has placed himself in one of the squares and does not move. Should the player select that particular square, he loses. If the player is successful in acquiring 1000 or more points, he wins and the Devil will reveal his hiding place to the player.

There are two random statements during the course of play. The initial RANDOM will set the computer selected randomizations to any possible number. The second encounter of the random instruction sets up another computer selection but within selected boundaries for the variables W and TT. This assures many totally unknown possibilities of where the Devil is hiding and the different point values which are also hiding.

A basic description of the program follows.

LINE DESCRIPTIONS:

10 Defines variables and strings. A GOSUB 1000 checks for the end of the program indicating a proper CLOAD of the tape. The game instructions are located there.
20 After printing the introductions to the game, the player is asked if he would like a hard game.
30 The INKEY$ function acts as a switch, with variable UU being the switch. If the answer to line 20 is YES then the logic ON is a 1 so that UU=1. For NO, a logical 0 is stored in UU. Only a Y or an N can be INKEYed.
50 RANDOM sets up the first or primary randomization of variables W and TT to assure total random selection of both values. Winnings tally MN is set to zero for a replay of the game. The initial resetting of all variables is accomplished by the RUN and CLEAR instructions.
100 The READ instruction has 10 variables assigned to it. Variable K is the test for a match comparison with variable W. ZA thru ZI are game square values read from the DATA bank.
105 Test variable K against W. This sets up a loop, so when they do not match, the program reverts back to line 100 until a match is found. W is the RANDOM selection for 1 of 18 possibilities. To alter the number of DATA lines, the RND(18) in line 60 must be altered to reflect the actual number of DATA lines.
110 - 280 DATA bank. Each line has 10 values, with the first one being the test value K.
290 RESTORE is executed once K=W so that the next round will not hang-up.
300 - 330 Draws playing board. Line 330 places numbers in the 9 boxes.
410 Variable TT now goes into a second RANDOM for absolute randomization. Only an integer between 1 and 9 inclusive can be selected for play.
420 - 500 Sets values of variable DV, which is where the Devil will be printed. Once the value is set, the program will drop to line 510.
510 - 600 Prints tallies on screen.
560, 610, 620 Variable E is used as a switch. Rather than use the logical 1 and 0 for YES and NO, the actual number of players is used for ease of reference, so that a 1 or 2 is used here.
590 Variables TL(sum total), PL(player #1), PY(player #2), and DL(Devil) are used to tally

the games played.

610 - 620 Tests for players turns. If player #1 is playing, a right arrow will be printed to the left of that players turn. Sometimes, when the Devil is found on the first turn of the round, the switch will remain set so that player may have another chance. In this case the variable SW is the switch.

630 Tests for winning the round. If a player accumulates 1000 or more points, he wins.

640 During play, this asks the player to CHOOSE A SQUARE.

650 The INKEY$ function limits the players' choices to integers 1 thru 9.

660 Prints players' choice of square.

670 This tests selection S against the hidden Devil TT and if they equal, the round is over and the player lost.

680 - 760 Will print the variables ZA to ZI in the selected square, from the data selection if the Devil is not there. ZA will always go in square 1, ZB in square 2, ZC in square 3, and so on. MN is the cumulative register for winnings of the round. The GOTO 530 is a loop type instruction to let the player resume playing as he has been successful so far in avoiding the Devil.

770 Prevents an accidental fall-through into the ending routines and doubles as a GOTO for line 760.

780 - 870 Print Devil and loss statements.

790 - 800 Switches UU and SW are tested. If a hard game (UU=1) and player #1 (SW=0) or player #2 (SW=1) then points are deducted from that players' grand total.

810 MN is set to zero.

820 FY is set to zero to print loss statement.

830 - 850 Prints Devil at location selected in lines 420 - 500.

860 If player has won the round, FY is turned on or FY=1, so that the loss statement will not be printed.

870 Player loss statement and Devil win tally (DL).

880 Asks to play again.

890 - 900 For 2 players, this routine changes players turns. The GOTO in 890 prevents SW from returning to zero once it is set to 1.

910 This loops into the print Devil routine upon a player winning.

920 - 940 INKEY$ to either play again or stop.

950 Should anything other than a Y or an N be INKEYed, then this loops the program back to line 910.

960 Prints player winning statement.

970 - 980 Calculates the players' winnings.

990 Sets switch FY to 1 to print where the Devil was hiding.

1000 - 1190 Prints game instructions and starting routine.

1200 - 1260 Sets up play for either 1 or 2 players. For 1 player, it is YOU, and for 2 players, you enter their names.

PROGRAM LISTING:

```
1Ø CLS:CLEAR:DEFINTA-Z:DEFSNG G,V,I:GOSUB1ØØØ
2Ø PRINT CHR$(23):PRINT:PRINT:PRINT"WANT A HARD GAME?  (Y/N)"
3Ø Q$=INKEY$:IFQ$="" THEN 3Ø ELSE IF Q$="Y" THEN UU=1 ELSE IF Q$="N" THEN UU=Ø ELSE 3Ø
4Ø 'BEAT THE DEVIL GAME DEVELOPED 12/6/79 BY JOHN R. WARRICK, P.O. BOX 82 BREWSTER STREET
  , BROWNS MILLS, NEW JERSEY Ø8Ø15
5Ø CLS:MN=Ø:RANDOM
6Ø W=RND(18):IF W=Ø THEN 6Ø
1ØØ READ K,ZA,ZB,ZC,ZD,ZE,ZF,ZG,ZH,ZI
1Ø5 IF K.NE.W THEN 1ØØ
11Ø DATA 1,5ØØ,45Ø,2ØØ,5Ø,15Ø,1ØØ,25Ø,3ØØ,35Ø
12Ø DATA 2,4ØØ,45Ø,5ØØ,5Ø,1ØØ,25Ø,3ØØ,2ØØ,35Ø
13Ø DATA 3,5Ø,1ØØ,2ØØ,5ØØ,4ØØ,45Ø,3ØØ,25Ø,35Ø
14Ø DATA 4,1ØØ,2ØØ,5ØØ,3ØØ,25Ø,15Ø,35Ø,45Ø,4ØØ
15Ø DATA 5,5Ø,1ØØ,15Ø,5ØØ,35Ø,25Ø,4ØØ,45Ø,2ØØ
```

```
16Ø DATA 6,1ØØ,5ØØ,45Ø,35Ø,4ØØ,5Ø,15Ø,25Ø,2ØØ
17Ø DATA 7,5ØØ,5Ø,15Ø,25Ø,4ØØ,45Ø,35Ø,3ØØ,2ØØ
18Ø DATA 8,25Ø,1ØØ,5Ø,3ØØ,15Ø,45Ø,25Ø,2ØØ,5ØØ
19Ø DATA 9,35Ø,45Ø,4ØØ,5Ø,1ØØ,15Ø,5ØØ,25Ø,35Ø
2ØØ DATA 1Ø,35Ø,45Ø,25Ø,15Ø,5Ø,5ØØ,1ØØ,3ØØ,4ØØ
21Ø DATA 11,1ØØ,2ØØ,3ØØ,4ØØ,5ØØ,5Ø,15Ø,25Ø,35Ø
22Ø DATA 12,15Ø,5Ø,5ØØ,2ØØ,3ØØ,5ØØ,4ØØ,3ØØ,5Ø
23Ø DATA 13,5Ø,15Ø,5ØØ,4ØØ,5Ø,45Ø,25Ø,25Ø,4ØØ
24Ø DATA 14,1ØØ,15Ø,25Ø,5ØØ,4ØØ,3ØØ,35Ø,45Ø,25Ø
25Ø DATA 15,3ØØ,15Ø,15Ø,5Ø,5ØØ,45Ø,4ØØ,35Ø,4ØØ
26Ø DATA 16,5Ø,15Ø,1ØØ,5ØØ,3ØØ,45Ø,25Ø,25Ø,4ØØ
27Ø DATA 17,15Ø,45Ø,5Ø,5ØØ,35Ø,3ØØ,25Ø,2ØØ,5Ø
28Ø DATA 18,5ØØ,25Ø,2ØØ,5Ø,1ØØ,15Ø,4ØØ,35Ø,45Ø
29Ø RESTORE
3ØØ 'GAME BOARD ROUTINE
31Ø Y=3:FORX=2TO5Ø:SET(X,Y):NEXT:Y=12:FORX=2TO5Ø:SET(X,Y):NEXT:Y=21:FORX=2TO5Ø:SET(X,Y):
    NEXT:Y=3Ø:FORX=2TO5Ø:SET(X,Y):NEXT
32Ø X=2:FORY=3TO3Ø:SET(X,Y):NEXT:X=18:FORY=3TO3Ø:SET(X,Y):NEXT:X=34:FORY=3TO3Ø:SET(X,Y):
    NEXT:X=5Ø:FORY=3TO3Ø:SET(X,Y):NEXT
33Ø PRINT@199,"1";:PRINT@2Ø7,"2";:PRINT@215,"3";:PRINT@391,"4";:PRINT@399,"5";:PRINT@4Ø7
    ,"6";:PRINT@583,"7";:PRINT@591,"8";:PRINT@599,"9";
4ØØ 'TIC TAC TALLY AND BEAT THE DEVIL ROUTINE
41Ø TT=RND(9):IFTT.LE.Ø OR TT.GT.9 THEN 41Ø ELSE 42Ø
42Ø IFTT=1 THEN DV=67:GOTO51Ø
43Ø IFTT=2 THEN DV=75:GOTO51Ø
44Ø IFTT=3 THEN DV=83:GOTO51Ø
45Ø IFTT=4 THEN DV=259:GOTO51Ø
46Ø IFTT=5 THEN DV=267:GOTO51Ø
47Ø IFTT=6 THEN DV=275:GOTO51Ø
48Ø IFTT=7 THEN DV=451:GOTO51Ø
49Ø IFTT=8 THEN DV=459:GOTO51Ø
5ØØ IFTT=9 THEN DV=467
51Ø PRINT @41,"T O T A L S";
52Ø PRINT @94,"THIS ROUND";
53Ø PRINT @1Ø7," $";MN;
54Ø PRINT @224,"NAME         GAMES    WINNINGS";
55Ø PRINT @288,F$;:PRINT @3ØØ,PL;:PRINT @3Ø7,"  $";G;
56Ø IF E=1 THEN 58Ø
57Ø PRINT @352,J$;:PRINT @364,PY;:PRINT @371,"  $";V;
58Ø PRINT @416,"D E V I L";:PRINT @428,DL;:IF UU=1 THEN PRINT @435,"  $";I;
59Ø TL=PL+PY+DL
6ØØ PRINT @48Ø,"  TOTAL      ";TL;
61Ø IF E=2 AND SW=Ø THEN PRINT @284,CHR$(94);
62Ø IF E=2 AND SW=1 THEN PRINT @348,CHR$(94);
63Ø IF MN.GE.1ØØØ THEN 96Ø
64Ø PRINT @67Ø,"CHOOSE A SQUARE";
65Ø S$=INKEY$:IFS$=""OR S$.LE."Ø" OR S$.GT."9" THEN 65Ø
66Ø PRINT @687,".LT. ";S$;" .GT.";
67Ø S=VAL(S$):IF S=TT THEN 78Ø
68Ø IF S=1 THEN PRINT @131,ZA;:MN=MN+ZA:GOTO 53Ø
69Ø IF S=2 THEN PRINT @139,ZB;:MN=MN+ZB:GOTO 53Ø
7ØØ IF S=3 THEN PRINT @147,ZC;:MN=MN+ZC:GOTO 53Ø
71Ø IF S=4 THEN PRINT @323,ZD;:MN=MN+ZD:GOTO 53Ø
72Ø IF S=5 THEN PRINT @331,ZE;:MN=MN+ZE:GOTO 53Ø
73Ø IF S=6 THEN PRINT @339,ZF;:MN=MN+ZF:GOTO 53Ø
74Ø IF S=7 THEN PRINT @515,ZG;:MN=MN+ZG:GOTO 53Ø
75Ø IF S=8 THEN PRINT @523,ZH;:MN=MN+ZH:GOTO 53Ø
```

```
76Ø PRINT @531,ZI;:MN=MN+ZI
77Ø GOTO 53Ø
78Ø 'PRINTS DEVIL
79Ø IF UU=1 AND SW=Ø THEN G=G-MN:I=I+MN
8ØØ IF UU=1 AND SW=1 THEN V=V-MN:I=I+MN
81Ø MN=Ø
82Ø FY=Ø
83Ø PRINT @DV,CHR$(147);STRING$(2,131);CHR$(163);
84Ø PRINT @DV+64,CHR$(183);CHR$(187);CHR$(183);CHR$(187);
85Ø PRINT @DV+128,CHR$(32);STRING$(2,143);
86Ø IFFY=1 THEN FY=Ø:GOTO 92Ø
87Ø DL=DL+1:PRINT @835,"   Y   O   U     L   O   S   E   !   !   !"
88Ø PRINT @91Ø,"CARE TO TRY AGAIN ? (Y/N)"
89Ø IF E=2 AND SW=Ø THEN SW=1:GOTO 91Ø
9ØØ IF E=2 AND SW=1 THEN SW=Ø
91Ø IF FY=1 GOTO 83Ø
92Ø A$=INKEY$:IF A$="" THEN 92Ø
93Ø IF A$="Y" THEN 5Ø
94Ø IF A$="N" THEN PRINT @91Ø,"IF YOU THINK I'M GONNA PAY OFF, YOU'RE CRAZY!":END
95Ø GOTO 88Ø
96Ø PRINT @835,"Y O U   B E A T   T H E   D E V I L   A N D   W O N  ! ! !"
97Ø IF E=1 OR (E=2 and SW=Ø) THEN G=G+MN:PL=PL+1
98Ø IF E=2 AND SW=1 THEN V=V+MN:PY=PY+1
99Ø FY=1:GOTO 78Ø
1ØØØ CLS:PRINTCHR$(23)
1Ø1Ø PRINT"   TRY TO BEAT THE DEVIL"
1Ø2Ø PRINT:PRINT"TEST YOUR E.S.P. BY GUESSING"
1Ø3Ø PRINT"WHICH SQUARE THE DEVIL IS"
1Ø4Ø PRINT"NOT IN. YOU ACCUMULATE A SCORE"
1Ø5Ø PRINT"FOR EACH TIME YOU AVOID THE"
1Ø6Ø PRINT"DEVIL IN A GAME."
1Ø7Ø PRINT"WHEN THE DEVIL BEATS YOU, YOU"
1Ø8Ø PRINT"LOSE YOUR SCORE FOR THAT GAME."
1Ø9Ø PRINT"IN A HARD GAME, YOU LOSE WHAT"
11ØØ PRINT"SCORE YOU ACCUMULATE FROM YOUR"
111Ø PRINT"TOTAL WINNINGS AS WELL!"
112Ø PRINT:INPUT"**PRESS ENTER TO CONTINUE**";A$:CLS
113Ø PRINTCHR$(23):PRINT:PRINT"AT THE END OF EACH ROUND"
114Ø PRINT"THE DEVIL WILL SHOW UP IN THE"
115Ø PRINT"SQUARE WHERE HE WAS HIDING,"
116Ø PRINT"UNLESS, YOU FIND HIM FIRST!"
117Ø PRINT:PRINT"GOOD LUCK!!!"
118Ø PRINT"     GOOD E.S.P. !!!"
119Ø PRINT:INPUT"**PRESS ENTER TO CONTINUE**";A$:CLS
12ØØ PRINTCHR$(23):PRINT:PRINT"1 OR 2 PLAYERS?"
121Ø E$=INKEY$:IF E$="" OR E$.LT."1" OR E$.GT."2" THEN 121Ø
122Ø E=VAL(E$)
123Ø IF E=1 THEN F$="YOU       ":J$="          ":CLS:RETURN
124Ø PRINT:INPUT"ENTER FIRST NAME";F$
125Ø PRINT:INPUT"ENTER SECOND NAME";J$
126Ø CLS:RETURN
```

....

COMING IN MAY

In May, we will take an indepth look at how your TRS-80 functions. We will begin a series of tutorial articles which will take you from the novice's first look at Level I BASIC up through Level 2 and Disk BASIC, and finishing with techniques in programming and Assembly Language programming. Also, an intriguing simulation of our Solar System based on the 1982 "Grand Alignment" of all the planets in the Solar System.

SPECIAL!

TRS-80 USERS - TREAT YOUR COMPUTER TO THESE EXCITING PROGRAMS!

ENHANCED GRAPHICS AND MORE OPTIONS IN OUR CURRENT VERSIONS!

PRICED AS LOW AS 31¢ PER PROGRAM!

AIOS/1 - This program combines NATURAL LANGUAGE with action verb programming. 4K & up

NLOS/1 - Give your TRS-80 the power to read and understand ENGLISH! Build conversational data bases - solve problems and answer questions relating to information learned. 16K (NLOS/2 - 32K, enhanced, performs learned tasks, built-in vocabulary!)

MAZE/1 - Randomly generate and solve MAZES of selected complexity. 4K

CONSTELLATION - Unique graphics display the night sky, then travel to any star and view the night sky of that ALIEN planet. 16K (version 2, enhanced, built-in star chart!)

YG/1 - Players may challenge the TRS-80 to a game of YAHTZEE. 16K

CARTOON - Create and run ANIMATED PICTURES on your screen. 4K

CP/1 - Randomly generate and solve CROSSWORD PUZZLES - graphics. 16K

BGSG/1 - Command Colonial or Cylon fleets in BATTLESTAR GALACTICA. 16K

CHECKERS - Challenge your TRS-80 to a game of CHECKERS - graphics. 16K

LND/1 - Buy and manage properties as you build your REAL ESTATE empire. But beware, your tenants may give you trouble! 4K

MNP/1 - Challenge your TRS-80 to a game of MONOPOLY. 16K

SWG/1 - Challenge your TRS-80 to a CROSSWORD-like game. 4K

TRIVIA - Test your memory with this BRAIN TEASING game. 16K

POKER - Challenge your TRS-80 to a game of POKER. 4K

BWL/1 - Challenge your TRS-80 to a BOWLING match. 4K

CAL/1 - Turn your TRS-80 into a powerful CALCULATOR. 4K

TMT/1 - Chase a madman forward and backward in TIME. 4K

CLUE - Become a master DETECTIVE and solve murder mysteries. 4K

NB/1 - Hunt down the enemy's fleet in this exciting NAVAL BATTLE. 4K

AR/1 - Test your skill in a wrecked car in DEMOLITION DERBY. 4K

BB/1 - Challenge your TRS-80 to a BASEBALL game. 4K

CS/1 - Manage a nuclear power plant in CHINA SYNDROME. 4K

LL/1 - Attempt to land an extraterrestrial SPACE CRAFT. 4K

ENV/1 - Test your knowledge of ECOLOGY. 4K

RBT/1 - Guide a series of bombs in an attempt to blow up a group of invading ANDROIDS. 4K

WT/1 - Lead a WAGON TRAIN safely across the prairie. 4K

WAR/1 - Command airplanes, TANKS and ARMIES in a war. 4K

MS/1 - Test your MATH SKILLS at various levels of complexity. 4K

SHT/1 - Turn your TRS-80 into a SHOOTING GALLERY. 4K

PB/1 - Turn your TRS-80 into a PINBALL MACHINE. 4K

SW/1 - Engage in INTERSTELLAR CONFLICT against the Zetars. 16K

BNG/1 - Play an exciting game of BINGO with your TRS-80. 4K

GR/1 - Challenge your TRS-80 to a game of GIN RUMMY. 4K

BJ/1 - Challenge your TRS-80 to a game of BLACKJACK. 4K

PP/1 - Challenge your TRS-80 to a PING-PONG game. 4K

FOUR IN A ROW - Play the game of CONNECT FOUR - animated graphics. 16K

TWP/1 - Turn your TRS-80 into a powerful WORD PROCESSOR. 16K

TRG/1 - GENERATE REPORTS from tape data files with headings, control breaks, record selection and totals. 16K

TRS/1 - Multi-key field SORT UTILITY for tape data files. 16K

TDB/1 - Create, maintain and inquiry to TAPE DATA BASE files. 16K

INVASION - Prevent an ALIEN INVADER from destroying the Earth. 16K

LEVEL 2 BASIC! COMPLETE INSTRUCTIONS! CLOAD TESTED CASSETTES: $3.49! PROGRAM LISTINGS: $1.25! ORDER 2 OR MORE PROGRAMS - GET 10% OFF! ALL 41 PROGRAM LISTINGS FOR $12.95! PROGRAM LISTINGS EASILY ADAPTED TO PET, APPLE AND OTHER BASIC COMPUTERS!

SEND CHECK OR MONEY ORDER TO:

CYBERMATE * R.D.#3 BOX 192A * NAZARETH, PA 18064