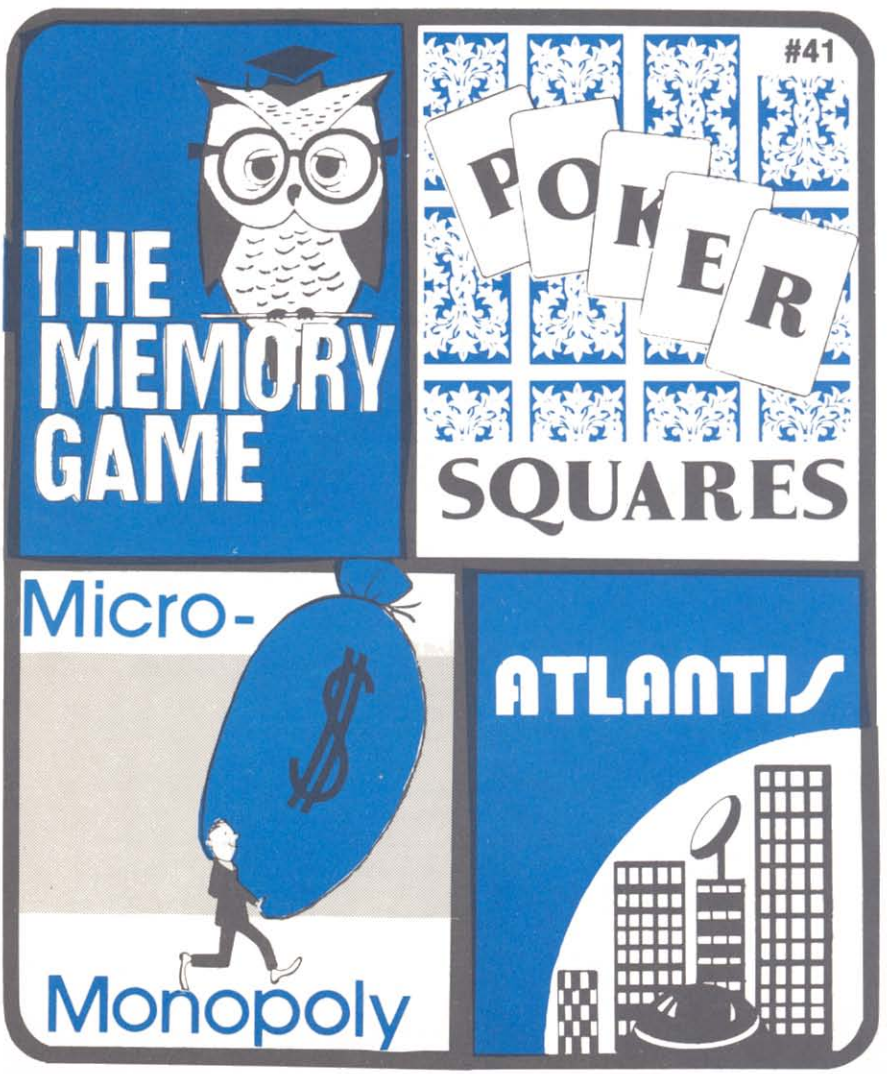
SoftSide™ Selections

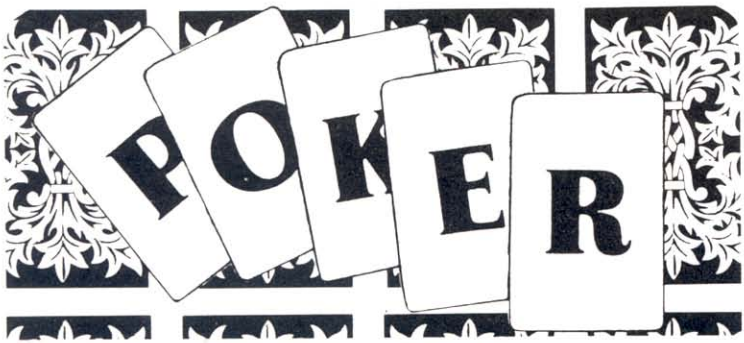


A product of **SoftSide** Publications, Inc.

6 South Street, Milford, NH 03055.

SoftSide™ Selections

Index



SQUARES

by Ron O'Laughlin and Bob Whitworth
TRS-80® version by Rich Bouchard

Poker Squares is a card game for one to four players for a TRS-80 Model I or III with 16K RAM (32K with disk).

The clock is ticking, and to get the best score, you must find the best location for the two of hearts you have just drawn. It fits fine in the first row, completing your two pairs, but wait — it spoils the straight in clubs in the second column. Finding the optimum placement for good cards and bad is the basic strategy of *Poker Squares,* an adaptation of the old poker solitaire card game.

The computer is the dealer — it shuffles the cards and turns up the first 25 cards, one at a time. Each player then drops the cards on a 5 x 5 grid by pressing one of the letters A to Y corresponding to the small rectangles that make up the grid. As the exposed top card on the deck is played onto the grid, the next card is turned up. After all 25 cards are on the grid, the computer adds and displays the scores.

The object of the game is to create ten poker hands, five across and five vertically, of five cards each. Diagonals do not count. Hands earn a score from zero to 30, depending on the relative difficulty of achieving that particular hand. (See the end of this article for a scoring chart and short tutorial on the ranking of poker hands).

Strategy is very important in this game. For example, you'll learn to go for flushes in one direction and straights, full houses, and so on, in the other. You will develop your own particular strategies and may actually become more proficient the next time your local poker group gets together.

You can play the game as solitaire, or up to four players can compete head-to-head. In the multiple-player mode, each person plays, in turn, the same 25 cards turned up in the same order. Dropping the cards into the grid in the best possible arrangement will reward you with the highest score. Don't peek at the competition! Each player has a set time to play each card, and may select his own time limit, providing a very effective handicapping system.

Beware of the following hazards: Every time the clock runs down to zero, the player receives a five-point penalty, and the clock is reset. If you hit an improper key (one that does not match one of the A to Y grid squares, or one that has been previously selected), the appropriate error message appears. Once a card has been put onto the grid, you can't move or pick it up again.

To interrupt the game, hit the up arrow on your TRS-80. Hitting any key resumes the contest.

Scoring Chart

Straight Flush	30
Four of a Kind	16
Straight	12
Full House	10
Three of a Kind	6
Flush	5
Two Pair	3
One Pair	1

Poker Tutorial

The following hands are listed in ascending order. Any hand listed will beat all hands listed above it. Note that the odds are different in real poker from what they are in *Poker Squares,* so the ranking below is somewhat different than in the scoring chart above.

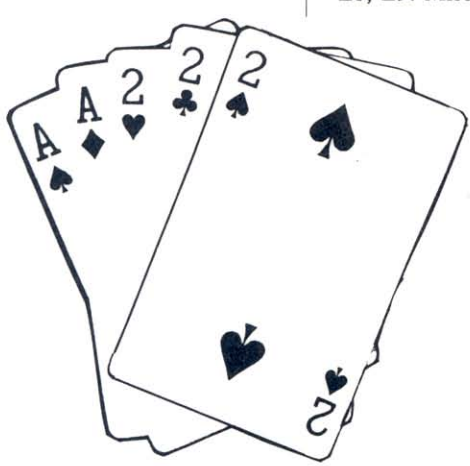
- NOTHING is just what it sounds like — no cards matching in any way for any score.
- ONE PAIR consists of any two cards of the same rank — such as two fives or nines.
- TWO PAIR consists of any two cards of the same rank, two cards of any other rank, and the fifth card unmatched.
- THREE OF A KIND is composed of any three cards of the same rank, with the other two cards in the hand unmatched.
- A STRAIGHT consists of any five cards of two or more different suits in sequence of rank. An ace may be used as part of a sequence either on the high or low end, such as in A-K-Q-J-10 or the A-2-3-4-5. The cards may appear in any order on the *Poker Squares* board.
- A FLUSH consists of any five cards of the same suit.
- A FULL HOUSE is composed of THREE OF A KIND along with a PAIR of another kind. Examples are as follows: 6-6-6-K-K and A-A-A-2-2.
- FOUR OF A KIND is next on the list. It consists of four cards of any one rank with the fifth card unmatched. Examples are: K-K-3-K-K and 3-3-3-3-10.
- A STRAIGHT FLUSH is the highest ranking hand and consists of five cards in sequence in the same suit. It is exactly as its name suggests — both straight and flush at the same time.

After you play *Poker Squares* a few times and observe the scoring routines, you will develop a strategy and start to get higher scores.

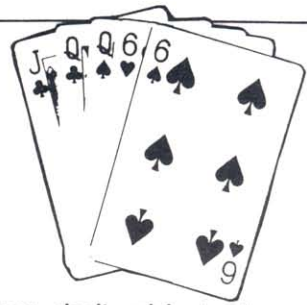
POKER SQUARES

Variables

A(*): Equal to 1 if card already dealt in round.
CL: Time remaining on clock.
CP(*): Seconds per card allowed for each player.
F1(*): Value of card dealt, from ace = 1 to king = 13.
F2(*): Suit of each card dealt, from 1 to 4.
G$: Graphic work string.
G$(*): Graphic strings for cards.
G4: Number of games played.
I$: Input for seconds per card.
K1: Equals 1 if there is a straight in the hand, 2 if there is a flush, or 3 if both.
K9: Number of cards in a straight for hand.
L(*,*): Card value at board position, from 0 to 13, with 0 = unoccupied to king = 13.
M(*,*): Suit at board position, from 1 to 4.
N: Number of players.
N$(*): Name of each player.
P: Score for each hand.
P7(*): Accumulated score per player.
P8(*): Total penalty points per player.
P9(*): Total score per player for the game.
Q, Q1: Used to help create graphic strings.
Q$: Single character typed during "keypress" routine.
S$(*): Number of cards in each suit for hand. Example: S(1) = 2 signifies that the hand contains two cards of suit one.
T(*): Number of cards by card value for hand. Example: T(*) = 2 signifies that the hand contains two cards of suit one.
U(*): Number of a kind in hand. Example: U(2) = 2 signifies that the hand contains two pairs.
V: Number of letter selected, A = 1, B = 2, etc.
V$: Character typed during game play.
VA$: List of one-character card values (A23...QK).
X, X1, X4, X5, X6, X7, X8: Miscellaneous uses.
XX: For-loop variable used in the subroutine that draws the game board.
Y: Random number (1-52) for card dealt.
Z1: Row number of letter selected on board.
Z2: Column number of letter selected on board.
Z8, Z9: Miscellaneous uses.



```
SS SS SS SS SS SS SS SS SS SS SS
SS                             SS
SS        TRS80 BASIC          SS
SS      'Poker Squares'        SS
SS Author: Ron O'Laughlin and SS
SS       Bob Whitworth         SS
SS Translator:  Rich Bouchard SS
SS     Copyright (c) 1983      SS
SS SoftSide Publications, Inc SS
SS                             SS
SS SS SS SS SS SS SS SS SS SS SS
```



If you don't wish to type this program, it is available on issue #41 SoftSide DV and CV.

Initialization and introduction.

```
100 RANDOM:CLEAR500:DEFINTA-Z:DIMN$(4),A(52),F1(25),F2(25),L(5,5
),M(5,5),P9(4),S(14),T(14),U(14),CP(4),P8(4),P7(4),G$(6)
110 CLS
120 PRINT@320,;:GOSUB1600
130 PRINT:PRINTTAB(16);"DO YOU WANT INSTRUCTIONS (Y/N) ";:GOSUB1
550:IFQ$="Y"THENGOSUB1260
```

Read the graphics data.

```
140 CLS:GOSUB1370
```

Zero the game scores.

```
150 G4=0
160 FORX=1TO4:P7(X)=0:NEXTX
170 GOTO270
```

Subroutine to zero the card codes.

```
180 FORX6=1TO14:S(X6)=0:T(X6)=0:U(X6)=0:NEXTX6
190 RETURN
```

Zero the scores and card-usage flags from previous game.

```
200 FORX=1TO4:P9(X)=0:P8(X)=0:NEXTX
210 FORX=1TO52:A(X)=0:NEXTX
```

Select 25 different cards, each one with a number from 1 to 52, and convert this number into a suit and pip value.

```
220 FORX=1TO25
230 Y=RND(52):IFA(Y)=1THEN230
240 A(Y)=1:F2(X)=INT((Y-1)/13)+1:F1(X)=Y-(F2(X)-1)*13
250 NEXTX
260 RETURN
```

Input the number of players.

```
270 CLS:GOSUB1600
280 PRINT@256,CHR$(30);"PLEASE ENTER NUMBER OF PLAYERS (1-4) ";T
AB(44);"? ";:GOSUB1550:N=VAL(Q$)
290 IFN<1ORN>4THENPRINT:PRINT"TRY AGAIN!";:GOTO280
```

Input each player's name.

```
300 FORX=1TON
```

POKER SQUARES

```
310 PRINT@320+X%64,CHR$(30);"NAME OF PLAYER";X;TAB(44);:INPUTN$(
X):IFN$(X)=""THEN310
320 N$(X)=LEFT$(N$(X),16):NEXTX
```

Input each player's time limit.

```
330 FORX=1TON
340 I$="":PRINT@384+(N+X)%64,CHR$(30);"SECONDS PER CARD FOR ";N$
(X);" (5-60)";TAB(44);:INPUTI$:CP(X)=VAL(I$)
350 IFCP(X)>60ORCP(X)<5THENPRINT"TRY AGAIN!":GOTO340
360 NEXTX
```

Start play for each player. Draw game board and prompt user to start.

```
370 GOSUB200
380 FORX=1TON
390 FORX4=1TO5
400 FORX5=1TO5:L(X4,X5)=0:M(X4,X5)=0
410 NEXTX5,X4
420 GOSUB1500:PRINT@960,"START OF PLAY FOR ";N$(X);" -- PRESS AN
Y KEY";
430 GOSUB1550:PRINTCHR$(29);CHR$(30);
440 FORX7=428TO556STEP64:PRINT@X7,STRING$(15,191);:NEXTX7
```

Main game loop.

```
450 FORX1=1TO25
```

Draw the "turned up" card.

```
460 PRINT@435,MID$(VA$,F1(X1),1);G$(F2(X1));
470 PRINT@684,"PLACE YOUR CARD";:CL=CP(X):PRINT@748,"TIME LEFT:"
;CL;"  ";
```

Check if user has decided where to place card.

```
480 FORZ8=1TO30:V$=INKEY$:IFV$=""THENNEXTZ8ELSE540
```

If no decision, decrement the clock.

```
490 CL=CL-1:PRINT@758,CHR$(30);CL;
500 IFCL>0THEN480
```

Display message and assign penalty points when time runs out.

```
510 PRINT@759,"NONE";:P8(X)=P8(X)-5
520 FORZ9=1TO100:V$=INKEY$:IFV$=""THENNEXTZ9ELSE540
530 GOTO470
```

Check for up arrow key to pause the game.

```
540 IFV$=CHR$(91)THENGOSUB1230:GOTO490
```

Check for permissible card drop value.

```
550 IFV$<"A"ORV$>"Y"THEN480
```

Adjust card drop value to the range 1-25, and convert it to a row and column.

```
560 V=ASC(V$)-64
570 Z1=INT((V-1)/5)+1
580 Z2=V-(Z1-1)%5
```

Check for card drop onto a previously used location.

```
590 IFL(Z1,Z2)<>0THENPRINT@876,"SQUARE TAKEN";:GOTO480
```

Decode the card location.

```
600 PRINT@876,CHR$(30);:L(Z1,Z2)=F1(X1):M(Z1,Z2)=F2(X1)
```

Draw the card in the selected square.

```
610 PRINT@Z2*8+Z1*192-200,MID$(VA$,F1(X1),1);G$(F2(X1));
```

Continue for all 25 cards.

```
620 NEXTX1
```

Remove the "up" card after the 25th card is placed. Then calculate the score by column.

```
630 FORX7=424TO960STEP64:PRINT@X7,CHR$(30);:NEXTX7:PRINT@297,"PO
INTS ACROSS: ";:PRINT@364,;
640 FORX7=1TO5
650 GOSUB180
660 FORX8=1TO5
670 S(M(X7,X8))=S(M(X7,X8))+1:T(L(X7,X8))=T(L(X7,X8))+1
680 NEXTX8
690 GOSUB930
700 NEXTX7
710 PRINTCHR$(8);TAB(59);"=";P;:P9(X)=P9(X)+P:P=0
```

Calculate score by column.

```
720 PRINT@425,"POINTS DOWN:";:PRINT@492,;
730 FORX7=1TO5
740 GOSUB180
750 FORX8=1TO5
760 S(M(X8,X7))=S(M(X8,X7))+1:T(L(X8,X7))=T(L(X8,X7))+1
770 NEXTX8
780 GOSUB930
790 NEXTX7
800 PRINTCHR$(8);TAB(59);"=";P;:P9(X)=P9(X)+P:P=0
```

Display penalty points, and final score.

```
810 PRINT@553,"PENALTY POINTS";TAB(59);"=";P8(X);:P9(X)=P9(X)+P8
(X)
820 PRINT@617,"TOTAL SCORE";TAB(59);"=";P9(X);:P7(X)=P7(X)+P9(X)
```

Wait for a keypress.

```
830 PRINT@745,"ANY KEY TO CONTINUE ";:GOSUB1550
```

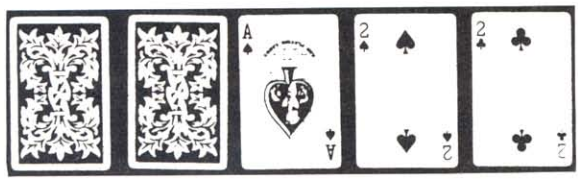
Continue for next player.

```
840 NEXTX
```

Display summary of scores.

```
850 CLS:GOSUB1600:PRINTTAB(22);"**  S U M M A R Y  **":G4=G4+1:P
RINT:PRINTTAB(23);"NUMBER OF GAMES =";G4
860 PRINT:PRINTTAB(13);"PLAYER     LAST     CUMULATIVE    AVERAG
E":PRINTTAB(13);" NAME      GAME       SCORE        SCORE":PRINT
TAB(7);STRING$(50,131)
870 FORX=1TON
```

POKER SQUARES



```
880 PRINTTAB(7);N$(X);TAB(24);P9(X);TAB(36);P7(X);TAB(49);INT((P
7(X)/G4)+.5)
890 NEXTX
```

Play another hand?

```
900 PRINT:PRINTTAB(20);"PLAY ANOTHER HAND (Y/N) ?";:GOSUB1550:IF
Q$<>"Y"THEN1610
910 CLS
920 GOTO330
930 FORX5=1TO13
940 IFT(X5)>=2THENU(T(X5))=U(T(X5))+1
950 NEXTX5
```

Check for four of a kind.

```
960 IFU(4)>=1THENPRINT"16+";:P=P+16:RETURN
```

Check for a full house or three of a kind.

```
970 IFU(3)<1THEN1000
980 IFU(2)>=1THENPRINT"10+";:P=P+10:RETURN
990 PRINT"6+";:P=P+6:RETURN
```

Check for number of pairs.

```
1000 IFU(2)=0THEN1030
1010 IFU(2)=2THENPRINT"3+";:P=P+3:RETURN
1020 PRINT"1+";:P=P+1:RETURN
```

Check for a flush.

```
1030 K1=0
1040 FORX5=1TO4
1050 IFS(X5)>=5THENK1=2
1060 NEXTX5
```

Check for a straight (ace high or low).

```
1070 K9=0:T(14)=T(1)
1080 FORX5=1TO14
1090 IFK9=5THEN1160
1100 IFK9>0THEN1130
1110 IFT(X5)=0THEN1160
1120 K9=1:GOTO1160
1130 IFT(X5)<>1THEN1150
1140 K9=K9+1:GOTO1160
1150 K9=0
1160 NEXTX5
```

```
1170 IFK9=5THENK1=K1+1
```
Check for a "nothing" hand.
```
1180 IFK1(=0THENPRINT"0+";:RETURN
```
Determine whether the hand is a straight, flush, or straight flush.
```
1190 ONK1GOTO1200,1210,1220
1200 PRINT"12+";:P=P+12:RETURN
1210 PRINT"5+";:P=P+5:RETURN
1220 PRINT"30+";:P=P+30:RETURN
```
Up arrow key time-out.
```
1230 PRINT@960,"HIT ANY KEY TO RETURN TO GAME ";
1240 GOSUB1550:PRINTCHR$(29);CHR$(30);
1250 RETURN
```
Instructions.
```
1260 CLS:PRINTTAB(25);"INSTRUCTIONS"
1270 PRINT:PRINT"    THE TRS-80 DEALER WILL TURN UP 25 CARDS ONE
 AT A TIME.  YOU MUST SELECT ONE OF THE A-Y LETTERED WHITE SQUAR
ES ON WHICH TO"
1280 PRINT"DROP EACH CARD AS IT COMES UP.":PRINT:PRINT"    THE O
BJECT IS TO CREATE 10 POKER HANDS OF FIVE CARDS EACH."
1290 PRINT"THEY ARE THE 5 ROWS ACROSS AND THE 5 COLUMNS DOWN."
1300 PRINT:PRINT"    WHEN SEVERAL PLAYERS COMPETE, ALL WILL BE D
EALT THE SAME 25 CARDS.  HOW YOU PUT YOUR CARDS DOWN WILL DETERM
INE SCORING."
1310 GOSUB1540
1320 PRINT@128,CHR$(31);"    EACH HAND WILL EARN A SCORE OF 0 -
30 DEPENDING ON YOUR     SKILL AND SOME LUCK."
1330 PRINT:PRINT"BETTER PLAYERS MAY BE GIVEN A HANDICAP BY ALLOW
ING THEM LESS    TIME TO PLAY.  THE PENALTY IS 5 POINTS EVERY TI
ME THE TIMER"
1340 PRINT"REACHES ZERO.":PRINT:PRINT"    AFTER EACH ROUND AN AC
CUMULATIVE SCORE SUMMARY PAGE WILL BE DISPLAYED FOR EACH PLAYER
TO SEE JUST HOW HE ";CHR$(34);"STACKS UP";CHR$(34)
1350 PRINT:PRINT"    GOOD LUCK AND NO CHEATING!"
1360 GOSUB1540:RETURN
```
Create graphics strings.
```
1370 G$=CHR$(26)+STRING$(6,8)
1380 FORX=1TO4:READQ
1390 FORX6=1TOQ:READQ1
1400 IFQ1<>-1THENG$(X)=G$(X)+CHR$(Q1)ELSEG$(X)=G$(X)+G$
1410 NEXTX6,X
1420 DATA 20,191,159,135,139,175,191,-1,189,144,144,160,160,190,
-1,191,183,177,178,187,191
1430 DATA 20,191,191,135,139,191,191,-1,183,128,145,162,128,187,
-1,191,183,177,178,187,191
1440 DATA 20,131,131,187,147,131,171,-1,128,174,191,191,132,170,
-1,176,176,187,177,176,186
```

POKER SQUARES

```
1450 DATA 20,131,187,147,187,147,171,-1,130,175,191,191,135,170,
-1,176,176,187,177,176,186
1460 G$=G$+CHR$(8):G$(5)=STRING$(7,191)+G$+STRING$(7,191)+G$+STR
ING$(7,191)
1470 G$(6)=STRING$(8,32)+G$+STRING$(8,32)+G$+STRING$(8,32)
1480 VA$="A23456789TJQK"
1490 RETURN
```

Subroutine to draw the game board.

```
1500 CLS
1510 FORXX=0TO4:FORY=0TO4:PRINT@XX*8+Y*192,G$(5);:PRINT@XX*8+Y*1
92+66," ";CHR$(65+XX+Y*5);" ";:NEXTY,XX
1520 FORY=1TO4:PRINT@Y*192,;:FORXX=0TO4:PRINTSTRING$(7,188);CHR$
(25);:NEXTXX,Y
1530 RETURN
```

Subroutine to wait for a keypress. Entry at 1540 displays a message before getting the character, while an entry at 1550 simply waits for a keypress.

```
1540 PRINT@979,"HIT ANY KEY TO CONTINUE ";
1550 Q$=INKEY$
1560 PRINTCHR$(143);CHR$(24);:FORQ=1TO5:Q$=INKEY$:IFQ$=""THENNEX
TQ:PRINT" ";CHR$(24);:FORQ=1TO5:Q$=INKEY$:IFQ$=""THENNEXTQ:GOTO1
560
1570 IFQ$>=CHR$(97)ANDQ$<=CHR$(122)THENQ$=CHR$(ASC(Q$)-32)
1580 IFQ$>=" "ANDQ$<="Z"THENPRINTQ$;ELSEPRINT" ";
1590 RETURN
```

Print "Poker Squares" message in a graphic box.

```
1600 PRINTTAB(18);CHR$(188);STRING$(27,140);CHR$(188):PRINTTAB(1
8);CHR$(191);" P O K E R  S Q U A R E S ";CHR$(191):PRINTTAB(18
);STRING$(29,131):RETURN
```

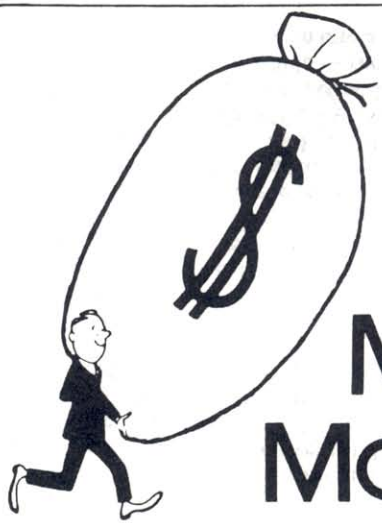
Program end.

```
32767 CLS:END
```

SWAT TABLE

For **TRS-80® POKER SQUARES**

LINES	SWAT CODE	LENGTH	LINES	SWAT CODE	LENGTH
100 - 210	HB	347	940 - 1050	FD	241
220 - 330	MN	343	1060 - 1170	JU	176
340 - 450	TQ	358	1180 - 1290	RZ	504
460 - 570	SJ	339	1300 - 1340	EG	539
580 - 690	ZJ	343	1350 - 1450	HJ	506
700 - 810	RI	297	1460 - 1570	NW	440
820 - 930	JO	472	1580 - 32767	UJ	149



Micro-Monopoly

by Charles W. Scott
TRS-80® version by Paul Schifferli

Micro-Monopoly is a game program for a 48K TRS-80 Model I or III. It is the bonus program on issue #41 SoftSide DV. See the coupon near the back of this booklet to order your disk.

Charles Darrow invented the board game *Monopoly* in the days of the Great Depression, and for five decades, *Monopoly* has allowed generations to play out their Everyman fantasies of being land tycoons in Atlantic City. *Micro-Monopoly* brings a two-player version of the game to your TRS-80, pitting you against the wily intelligence of the computer.

Micro-Monopoly follows the published rules, with a few minor exceptions chosen to make the two-player format more playable. The ten percent unmortgaging surcharge, required by the original rules, is strictly enforced, as is the rule prohibiting jailed players from collecting rent.

The game deviates from the official rules four times:

- The popular "Free Parking" pot is an option with *Micro-Monopoly*. Play this option the way you like it.
- Properties are never auctioned, as this is undesirable in a two-player game.
- Hotels appear on the various displays as five houses, although rent appropriate for hotels is charged.
- You may not trade properties with *Micro-Monopoly*. If you want to unload a property, you may sell it to the bank.

The operation of the game is explained in the instructions offered at the beginning of play, and all prompts are self-explanatory. So gather up your real-estate theories, and try to gain absolute dominion over Atlantic City.

DV BONUS

THE MEMORY GAME

by Rich Bouchard

The Memory Game is a one to eight player memory game for the TRS-80® Model I or III with 16K (cassette) or 32K (disk).

How sharp is your memory? Your computer challenges its powers in *The Memory Game.*

As the game opens, you are presented with a board of graphic blocks — nine, sixteen, 25 or 36 of them, depending on the difficulty level you choose. Each block is identified by a single letter or number. A number is hidden behind each block, ranging from one to the number of blocks on the screen, and the object of the game is to find every hidden number in sequential order. To see the number hidden behind a block, strike the keyboard character used to identify it. The number beneath the appropriate block then flashes briefly. If it is a one, the block disappears and the numeral "1" is displayed at the bottom of the screen in a small graphic rectangle. If it is not a one, the graphic block reappears, and the player has another chance to search for the "1".

After he's found it, the player must search for the two, then the three, and so on until he has found every hidden number. Take heart, however: the numbers remain stationary. That is, if the number "4" is behind block "B"at the start of the game, it remains there until the game is over. So, when the player needs to find the four, later in the game, he need only remember where it lies.

There are four difficulty levels, numbered from three to six. The levels reflect the number of columns and rows of blocks displayed. Difficulty level four, for example, has four rows of four blocks, for a total of sixteen blocks.

You may set another difficulty factor by entering a time limit for each number search. If you do not want a time limit, hit ENTER when asked for it. A player may then continue searching for numbers (almost) forever.

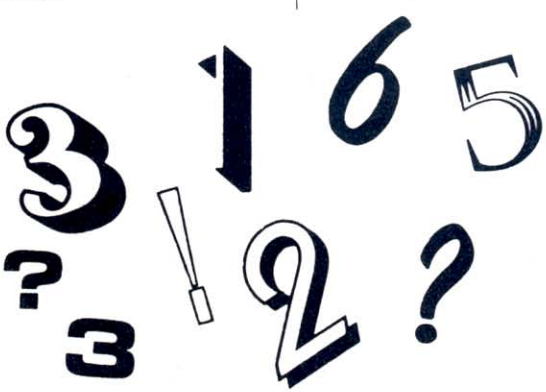
After the player has found all the hidden numbers, he is given a score based on the amount of time it took him/her to find all the numbers. This score is adjusted for the game's difficulty level. The highest possible scores are 100 at level three, 250 at level four, 500 at level five and 999 at level six. Scores this high are practically impossible — the chance of a player scoring 999 points at skill level six is less than one in one duodecillion (a "1" followed by 39 zeros)!

At the beginning of each multi-player game, *The Memory Game* asks "SAME PUZZLE FOR ALL PLAYERS?" If you answer "Y", the numbers hidden behind each block remain the same for every player. If you answer "N", the numbers are scrambled for each player. Generally, you will choose the "N" option. Sometimes, however, especially when players are competing with one another, you will want to enter a "Y" response. You can choose an interesting variation to this game by selecting a multi-player game, using the same puzzle for every player, and then playing every player's game yourself. In this variation, the knowledge gained concerning the position of the hidden numbers during your first game is valid during the other games. See how much this knowledge can help your score.

The Memory Game uses sound effects, so be sure to connect an amplifier to the AUX jack of your computer.

Variables

A,A$: Multi-purpose work variables.
A1,A2: Point to start of Machine Language sound routine.
B1(NS): List of numbers behind each graphic block at the start of a game.
B2(NS): Similar to B1(NS), but contains zeros for any block removed from play during current player's game.
C: Counter used to determine when TI should be incremented.
G1$,G2$: Graphic rectangles.
HP: Number of player with highest score for current game.
HS(NS), HS$(NS): High score and scorer's initials for each skill level.
ID$: A list of single character identifiers for each graphic block on the screen.
J: Work variable.
LV: Current skill level.
NB: Number of blocks in current board. NB is always equal to the square of LV.
NP: Number of players.
NS: Number of possible skill levels.
NU: Last number player has "found."
PB: Screen position of upper-left corner of playing board.
PL: Number of player whose turn it is.
PN: Screen position for display of variable NU.
S(NP): Scores for each player.
SP: -1 if all players are using the same board set-up; 0 if the numbers are to be scrambled for each player.



THE MEMORY GAME

```
SS SS SS SS SS SS SS SS SS SS SS
SS                             SS
SS         TRS-80 BASIC        SS
SS      'The Memory Game'      SS
SS   Author:  Rich Bouchard    SS
SS     Copyright (c) 1983      SS
SS SoftSide Publications, Inc  SS
SS                             SS
SS SS SS SS SS SS SS SS SS SS SS
```

If you don't wish to type this program, it is available on issue #41 SoftSide DV and CV.

Initialize sound routine, clear string space, and define all variables as integers.

```
100 CLS:GOSUB60000
110 CLEAR:CLEAR500:DEFINTA-Z:GOTO1000
```

Subroutine to get a single character into A$.

```
200 A$=INKEY$
210 PRINTCHR$(143);CHR$(24);:U=USR(RND(5)+512):FORT=1TO5:A$=INKE
Y$:IFA$=""THENNEXTT:PRINT" ";CHR$(24);:FORT=1TO5:A$=INKEY$:IFA$=
""THENNEXTT:GOTO200
220 IFA$>=" "THENPRINTA$;ELSEPRINT" ";
230 PRINTCHR$(24);:RETURN
```

Subroutine to get a 1-3 character string into A1$.

```
240 A1$="":A$=INKEY$
250 GOSUB210
260 IFA$=CHR$(13)THENIFA1$=""THENA1$="???":RETURNELSERETURN
270 IF(A$=CHR$(8)ORA$=CHR$(24))ANDA1$<>""THENPRINTCHR$(24);:A1$=
LEFT$(A1$,LEN(A1$)-1)
280 IFA$>=" "ANDLEN(A1$)<3THENPRINTCHR$(25);:A1$=A1$+A$
290 GOTO250
```

Initialize variables and arrays.

```
1000 GOSUB2900
```

Main program loop starts here.

Display title page and get skill level, number of players, and time limit.

```
1010 GOSUB2300
```

Initialize variables concerning the playing board size. Set current player variable (PL) to 1.

```
1020 GOSUB2600
1030 PL=1
```

Scramble and display playing board.

```
1040 GOSUB2700
```

Wait for user to strike a key. If a proper block is selected, display the number behind it. If "CLEAR" is hit, cancel game.

```
1050 A$=INKEY$:IFA$=CHR$(31)THENPRINT@320,"GAME CANCELED  ";:S(P
L)=0:FORT=1TO40:U=USR(T*256+4):NEXTT:GOTO1280
1060 IFA$=""ORA$>"Z"THENC=C+1:IFC<20THEN1050ELSE1120
1070 A=ASC(A$)-64:IFA>-16ANDA<-6THENA=A+42ELSEIFA=-16THENA=36
1080 IFA<10RA>NBORB2(-A*(A>0))=0THENU=USR(2810):GOTO1130
1090 Y=INT((A-1)/LV):X=A-Y*LV:PRINT@PB+Y*128+X*6-6,B2(A);
1100 IFB2(A)=NU+1THEN1170
1110 U=USR(5170):FORJ=1TO50:NEXTJ:PRINT@PB+Y*128+X*6-6,CHR$(191)
;MID$(ID$,A,1);CHR$(191);:GOTO1130
1120 U=USR(2570)
```

Increment timer and end game if out of time.

```
1130 C=0:TI=TI+1
1140 IFTL=0THENPRINT@330,TI;ELSEPRINT@330,TL-TI;
1150 IFTL=0ORTI<TLTHEN1050
1160 PRINT@320,"TIME EXPIRED!!";:U=USR(-1):S(PL)=0:GOTO1280
```

Player has found the next number. Display the number and erase the block it was behind. Continue if there are more numbers still hidden.

```
1170 NU=NU+1:PRINT@PN,NU;:U=USR(5140):U=USR(7700):U=USR(5140)
1180 B2(A)=0
1190 PRINT@PB+Y*128+X*6-6,STRING$(3,32);
1200 IFNU<NBTHEN1120
```

Player has found all the hidden numbers. Play a sequence of notes, then compute and display score.

```
1210 FORT=60TO160STEP10:A=T*256+T/4+65536*(T>127)
1220 U=USR(A):NEXTT
1230 FORT=160TO60STEP-10:A=T*256+T/4+65536*(T>127)
1240 U=USR(A):NEXTT
1250 IFLV=3THENS(PL)=108-TIELSEIFLV=4THENS(PL)=(181-TI)*1.5+1ELS
EIFLV=5THENS(PL)=(274-TI)*2ELSES(PL)=(368-TI)*3
1260 IFS(PL)<0THENS(PL)=0
1270 PRINT@320,"YOUR SCORE:";S(PL);
```

Wait until the player is done reading his or her score. Then either go on to the next player, or display the final scores and allow another game to be started.

```
1280 PRINT@19," HIT <ENTER> TO CONTINUE ";
1290 GOSUB200:IFA$<>CHR$(13)THEN1290
1300 IFPL<NPTHENPL=PL+1:GOTO1040
1310 GOSUB2000
1320 GOTO1010
```

End of main program loop.

THE MEMORY GAME

Subroutine to display final results. If a player has beaten the high score, get his or her initials.

```
2000 CLS
2010 PRINTTAB(24);CHR$(188);STRING$(15,140);CHR$(188)
2020 PRINTTAB(24);CHR$(191);" FINAL RESULTS ";CHR$(191)
2030 PRINTTAB(24);STRING$(17,131)
2040 PRINT
2050 PRINT"GAME LEVEL:";LV;
2060 IFHS(LV-2)<>0THENPRINTTAB(30);USING"PREVIOUS HIGH SCORE: ##
# BY %  %";HS(LV-2);HS$(LV-2);
2070 PRINT:PRINT
2080 IFNP=1THENPRINT"YOUR SCORE:";S(1):HP=1:GOTO2150
2090 PRINT"SCORES:";
2100 FORT=1TONP:PRINTTAB(10);USING"#.   ###";T;S(T)
2110 NEXTT
2120 HP=1:FORT=2TONP:IFS(T)>S(HP)THENHP=T
2130 NEXTT
2140 IFS(HP)<>0THENPRINT@320+30+HP*64,"OUR WINNER!";
2150 IFS(HP)<=HS(LV-2)THEN2180
2160 PRINT@908,"NEW HIGH SCORE!  ENTER YOUR INITIALS >";
2170 GOSUB240:HS$(LV-2)=A1$:HS(LV-2)=S(HP)
2180 PRINT@896,CHR$(30);TAB(20);"HIT <ENTER> TO CONTINUE ";
2190 GOSUB200:IFA$<>CHR$(13)THEN2190
2200 RETURN
```

Subroutine to display title page and high scores, and get number of players, skill level, and time limit.

```
2300 CLS
2310 PRINTTAB(22);CHR$(188);STRING$(17,140);CHR$(188)
2320 PRINTTAB(22);CHR$(191);" THE MEMORY GAME ";CHR$(191)
2330 PRINTTAB(22);STRING$(19,131)
2340 PRINT
2350 PRINTTAB(13);"-> WRITTEN BY RICHARD BOUCHARD JR. <-"
2360 PRINT:PRINT"BOARD SIZES:";
2370 FORT=1TONS:PRINTTAB(6+T*12);T+2;"X";T+2;:NEXTT
2380 PRINT:FORT=1TONS:PRINTTAB(5+T*12);"----------";:NEXTT
2390 PRINT:PRINT"HIGH SCORES:";
2400 FORT=1TONS:PRINTTAB(5+T*12);:IFHS(T)=0THENPRINT"  NONE";ELS
EPRINTUSING"### BY % %";HS(T);HS$(T);
2410 NEXTT:PRINT:PRINT
2420 IFNP=1THENPRINTUSING" LAST SCORE:     ### POINTS";S(1);
2430 IFNP<2THEN2460
2440 PRINT"LAST SCORES:";:FORT=1TONP:PRINTSTRING$(6,32);USING"#.
 ###";T;S(T);:IFT=4ANDNP>4THENPRINT:PRINTTAB(12);
2450 NEXTT
2460 PRINT@836,"HIT A NUMBER 3-6 TO PLAY AT A SELECTED DIFFICULT
Y LEVEL,":PRINTTAB(20);"OR <CLEAR> TO EXIT GAME >";
2470 GOSUB200
2480 IFA$=CHR$(31)THENPRINT@832,CHR$(31);:GOTO32767
```

```
2490 IFA$<"3"ORA$>"6"THEN2470
2500 LV=VAL(A$)
2510 PRINT@301+LV*12,G1$;
2520 PRINT@832,CHR$(31);TAB(20);"HOW MANY PLAYERS (1-8)? ";
2530 GOSUB200:NP=VAL(A$):IFNP<1ORNP>8THEN2530
2540 PRINT@896,CHR$(31);TAB(19);"TIME LIMIT (IF DESIRED) ?";:GOS
UB240:TL=VAL(A1$):IFTL=0ANDASC(A1$)<>48ANDASC(A1$)<>63THEN2540
2550 IFNP=1THEN2580
2560 PRINT@960,CHR$(31);TAB(18);"SAME PUZZLE FOR ALL PLAYERS? ";
2570 GOSUB200:IFA$="Y"THENSP=-1ELSEIFA$="N"THENSP=0ELSE2570
2580 RETURN
```

Subroutine to compute number of blocks on current board (NB) and to determine position for board on screen (PN).

```
2600 NB=LV*LV
2610 IFLV=6THENPB=80:PN=927ELSEIFLV=5THENPB=146:PN=862ELSEIFLV=4
THENPB=213:PN=798ELSEPB=280:PN=734
2620 RETURN
```

Scramble and display board, then wait for player to be ready to start.

```
2700 IFPL=1ORSP=0THENA$=LEFT$("789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVW
XYZ",NB):FORT=1TONB:J=RND(LEN(A$)):B1(T)=ASC(MID$(A$,J,1))-54:A$
=LEFT$(A$,J-1)+MID$(A$,J+1):NEXTT
2710 S(PL)=0
2720 T1=0:C=0:NU=0
2730 CLS
2740 FORY=1TOLV:FORX=1TOLV:PRINT@PB+Y*128+X*6-134,CHR$(191);MID$
(ID$,Y*LV+X-LV,1);CHR$(191);:NEXTX,Y
2750 PRINT@PN-65,G2$;
2760 PRINT@320,;:IFTL=0PRINT"     TIME: 0";ELSEPRINT"TIME LEFT:"
;TL;
2770 IFNP>1THENPRINT@448,"PLAYER #";PL;
2780 FORT=1TONB:B2(T)=B1(T):NEXTT
2790 PRINT@16,"HIT <ENTER> WHEN READY TO START ";
2800 GOSUB200:IFA$<>CHR$(13)THEN2800
2810 U=USR(10020):PRINT@16,CHR$(30);TAB(19);"HIT <CLEAR> TO CANC
EL GAME";
2820 RETURN
```

Subroutine to initialize variables and arrays.

```
2900 NS=4:DIMHS(NS),HS$(NS),B1(36),B2(36)
2910 ID$="ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890"
2920 G1$=CHR$(188)+STRING$(7,140)+CHR$(188)+CHR$(26)+STRING$(9,2
4)+CHR$(191)+STRING$(7,25)+CHR$(191)+CHR$(26)+STRING$(9,24)+STRI
NG$(9,131)+" "
2930 G2$=CHR$(188)+STRING$(4,140)+CHR$(188)+CHR$(26)+STRING$(6,2
4)+CHR$(191)+"    "+CHR$(191)+CHR$(26)+STRING$(6,24)+STRING$(6,1
31)
2940 RETURN
```

THE MEMORY GAME

Program termination.

```
32767 END
```

POKE sound routine into high memory.

```
60000 Y=234:X=255:POKE-1,0:IFPEEK(-1)<>0THENX=191:POKE-16385,0:I
FPEEK(-16385)<>0THENX=127
60010 POKE16562,X:POKE16561,Y:CLEAR50:A1=PEEK(16561)+2:A2=PEEK(1
6562):A=A1+A2*256:Z=A-1:FORX=1TO20:Z=Z+1:Z=Z+65536*(Z)32767)
60020 READY:Z1=Z1+Y:POKEZ,Y:NEXT
60030 IFZ1<>2700THENCLS:PRINT"DATA BASE ERROR IN LINES 60050-600
60. CHECK LISTING.":PRINT:LIST60050-60060ELSEIFPEEK(16396)=201TH
ENPOKE16526,A1:POKE16527,A2ELSECMD"T":DEFUSR=A1+(A2+256*(A2>127)
)*256:POKE14308,0
60040 U=USR(2827):GOTO110
60050 DATA205,127,10,69,62,1,211,255,16,254,69,175
60060 DATA211,255,16,254,37,32,240,201
```

SWAT TABLE

For TRS 80® THE MEMORY GAME

LINES	SWAT CODE	LENGTH
100 - 290	RN	349
1000 - 1110	PV	410
1120 - 1230	QQ	345
1240 - 2020	XG	349
2030 - 2140	TY	322
2150 - 2350	XT	356
2360 - 2470	HP	467
2480 - 2600	NP	404
2610 - 2790	LP	503
2800 - 60010	KK	545
60020 - 60060	SJ	286

Tired of Typing?

- Use And Enjoy These Programs Right Away!
- Avoid Worry About Typos!
- Get Additional Programs to Enjoy Without Typing.

Order this Issue's Programs on Disk or Cassette!

Send the coupon below to:

SoftSide Publications, Inc.
6 South Street
Milford, NH 03055

SoftSide CV SoftSide DV

SoftSide Selections

ORDER THIS ISSUE'S MEDIA

Having difficulty finding the time to input the programs from this issue of ***SoftSide* Selections**? Don't despair — the cassette and disk versions are still available! **And**, each DV (disk version) contains an additional program.

☐ **#41 DV $16.00** ☐ **#41 CV $8.00**

Disk Only: ☐ Apple® ☐ IBM® PC
Disk or Cassette: ☐ Atari® ☐ TRS-80® Mod. I ☐ TRS-80® Mod. III

Name _____

Address _____

City/State _____ Zip _____

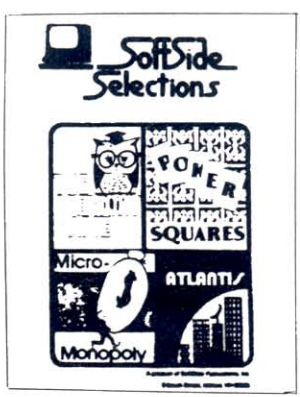
☐ **Check or Money Order**

☐ MasterCard ☐ VISA

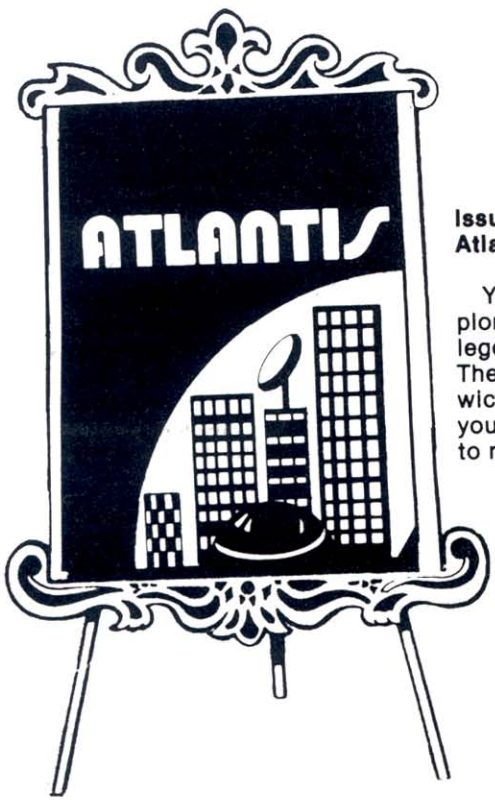
Name of Cardholder _____

MC# or Interbank#/VISA# _____

Exp. Date _____ Signature _____



SoftSide™ ADVENTURE SERIES



Issue 41 Adventure: Atlantis

You are an undersea explorer, and discover the legendary Lost Continent. The repressive regime of the wicked Prince Retep throws you in jail, and you resolve to rid Atlantis of this despot.

SoftSide Adventure Series CV DV

What would you say to a program that asks, "What do you want to do?" Well, you might say, "GET RUBY" or "KILL GIANT", because that's how the *SoftSide Adventure Series* works.

Each issue, the latest Adventure takes you to another world of fantasy, puzzles, and thrills. Your first task is often simple survival — and even that basic feat can be daunting until you figure out the *right* way to do it. You'll have to be ingenious and persevering, and your rewards will be great.

To "win" a fantasy/adventure game, you must solve the author's devious puzzles, and overcome the obstacles that confront you — whether they be dragons or desperadoes. Death, should it come, is transitory — just re-run the program to live again!

Experienced adventurers make detailed maps of each world as an aid to effecting a solution, but you can omit this exercise if your memory is exceptional. Express your wishes with one- or two-word commands, like "LOOK", "DROP KNIFE", or "GET RUBY". Use "I" to get an inventory of your possessions. The introduction to each Adventure explains this more fully.

To start up the Adventure, just run the program called "INTRO" or "INTRO/BAS" on your disk, or select the Adventure from the DV menu. On cassette, the INTRO program is the one just before the Adventure, which is the last program on the tape.

The adventure runs on any TRS-80 with at least 16K RAM (32K disk).

Here are the encrypted hints for *Volcano Island,* the adventure in Issue 40.

MZGREVH ORPV ZOXLSLO.

IFYYVI RH Z TLLW RMHFOZGLI.

DSZG WL ZM LXGLKFH ZMW Z GBKVDIRGVI IRYYLM SZEV RM XLNNLM?

GSV MZGREVH ZIV ZUIZRW LU GSV XZEV.

TIZHH ZMW YZNYLL NZPV Z TLLW GIZK.

HVZTFOO UVZGSVIH NZPV TLLW KVMH.

General Information

These are the standard procedures for the programs published in **SoftSide Selections**. Sometimes, a particular program does not lend itself to these procedures. Always read the specific instructions accompanying a program. They will instruct you if there are any variances from the following procedures. Also, back issues of **SoftSide** may differ in some details.

At the conclusion of each program listing in **SoftSide Selections,** we include a **SWAT (Strategic Weapon Against Typos)** Table. **SWAT** for the TRS-80 appeared in **SoftSide** Issue #30. If you missed Issue #30, we'll send you a free reprint of **SWAT**. Send a self-addressed, stamped envelope to:

SoftSide Publications, Inc.
Department **SWAT**
6 South Street
Milford, NH 03055

Please be sure to tell us that you have a TRS-80 computer.

Magnetic Media

Disks are available in Model I or Model III format. They contain the DOSPLUS operating system. A cover program runs automatically when you boot the disk. Back issues earlier than May 1981 are available only in Model I format. If you have a two-drive Model III, you can convert such disks with the CONVERT utility.

Tapes CLOAD in the normal manner on Model I's, and at low speed (500 baud) on Model III's. The first program is a cover/menu program; side two of the tape is a duplicate of side one.

SoftSide Selections disks and tapes are duplicated on reliable, professional equipment. Bad copies are exceedingly rare. Nevertheless, the trip through the mail occasionally results in damage to the sensitive magnetic media. If, after a reasonable number of attempts on well-adjusted, clean equipment, you are unable to load a program, return it to us along with an exact explanation of your problem. We will send you a replacement copy.

SoftSide Selections media are not copy protected. We urge you to make an archival backup copy of your disk or tape as soon as you receive it, as our replacement policy is valid only for 30 days. Please resist the urge to give away copies of copyrighted material.

Line Listings

The line listings in this booklet are in standard 64-column format, and they appear exactly as they should on your screen when you type LIST.

System Requirements

The necessary memory and other equipment you need to run a program are listed in the introductory paragraph of the article for each program. (Also see the **SoftSide Adventure Series** elsewhere in this booklet.)

The Best of SoftSide

NOW AVAILABLE ON DISK

☐ **YES**...I want to enjoy the best programs — Games, Utilities, Adventures — from SoftSide Magazine's past. Please rush my copies of The Best of SoftSide to:

Name _____

Address _____

City _____ State ____ Zip ____

Here's my order For The Best of SoftSide

Computer	Quantity Book Only $19.95 ea.	Quantity Book plus Disk $68.95 ea.
Apple®		
Atari®		
TRS-80® Mod I		
TRS-80® Mod III		
TOTAL COPIES	@$19.95	@$68.95

☐ I've already ordered The Best of SoftSide book. I'm enclosing $49 for the disks only.

Total Enclosed $_____

I'm paying by: ☐ Check ☐ Money Order
☐ VISA ☐ MasterCard

(Foreign order please include $5 postage PAY IN USA FUNDS ONLY)

VISA/MasterCard# and Interbank# _____ Exp. Date _____

Name of Cardholder _____

Signature _____

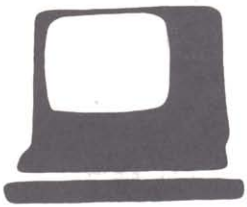
Like **SoftSide**? Wait 'til you've seen our **BEST!**

The Best of SoftSide!

For the past four years, **SoftSide** Magazine has been bringing Apple®, Atari®, and TRS-80® owners the best in BASIC software. But now you can do even better...**The Best of SoftSide.** From all our back issues, we've selected the most useful...the most entertaining...the most fun programs **SoftSide** has ever published.

The Best of SoftSide is available in three versions...one for Apple, Atari, and TRS-80. Each contains page after page of BASIC code for adventures, simulations, practical applications, and much more.

To order your copy of **The Best of SoftSide,** fill out the coupon, and mail it along with $19.95 to **SoftSide,** 6 South Street, Milford, New Hampshire 03055.



SoftSide™ Selections

Here's **SoftSide Selections,** the handy, pull-out booklet with program listings for your TRS-80 Model I or III computer. This issue, **SoftSide Selections** for the TRS-80 features:

- **Poker Squares** — This issue's Front Runner is a competitive graphic card game for one to four players.

- **The Memory Game** — Your TRS-80's RAM never forgets. How good is your memory?

- **TRS-80 DV Bonus Program: Micro-Monopoly** brings the popular board game to your TRS-80. Try to outwit its electronic real-estate savvy.

- **The SoftSide Adventure Series — Atlantis**, by Peter Kirsch.
Seek the lost city, and rid it of the evil Prince Retep.

SOFTSIDE PUBLICATIONS, INC.
Milford, New Hampshire