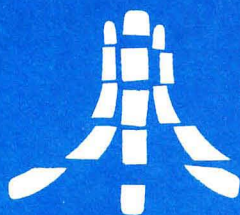


# SoftSide<sup>TM</sup> Selections



**Atari  
Blaster**



#41

**SQUARES**

**Five  
Card  
Stud  
Poker**



**ATLANTIS**



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

6 South Street, Milford, NH 03055.



# SoftSide<sup>TM</sup> Selections

## Index

---



### **FRONT RUNNER: Poker Squares**

by Ron O'Laughlin and Bob Whitworth  
Atari® version by Alan J. Zett

In this adaptation of the old poker solitaire game, the challenge is to find the optimum placement for both your good and bad cards .....

**2**



### **DV BONUS PROGRAM: Five Card Stud Poker** by Richard Spencer

Those arcade shoot-em-ups have nothing over the excitement of betting big bucks on the turn of a single card. In this game, you match wits with four computer-controlled players .....

**11**



### **Atari Blaster**

by Karl Haferer

The stakes are high in this game, as nasty aliens attempt to steal your most prized possession — your computer .....

**13**



### **ADVENTURE DISK and CASSETTE BONUS:**

#### **SoftSide Adventure Series: Atlantis**

by Peter Kirsch

Your undersea explorations lead you to the long-hidden city of Atlantis ruled by the evil King Retep. You'll need all your wits about you to survive your encounter with this malevolent despot .....

**20**

### **General Information**

and System Requirements .....


**22**

---

This booklet prepared by the staff of SoftSide Publications, Inc. Entire contents copyright © 1983 by SoftSide Publications, Inc., 6 South Street, Milford, NH 03055

ATARI® is a registered trademark of Warner Communications.





# POKER SQUARES

## Poker Squares

by Ron O'Laughlin and Bob Whitworth

Atari® version by Alan J. Zett

**Poker Squares is a card game for one to four players for an Atari 400/800/1200 with 32K RAM.**

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 four players can compete head-to-head. In the multiple-player mode, each person plays, in turn, the same 25 cards turned up into the same order. Dropping the cards in 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 CTRL-1 to halt play. Hitting CTRL-1 again resumes the contest.

Because of screen-size limitations, the Atari version does not prompt you to press the space bar between rounds.

### 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.

XFR\$, which appears in line 1800, contains a machine-language routine to set up a redefined character set. Here are explicit instructions to represent this string in the listing.

## POKER SQUARES

Each key is represented by something between brackets. Thus [h] means to type a lower-case "h"; likewise, [Atari] means to press the Atari logo key, which is found next to the right-hand shift key, and [CTRL-N] means to press the "N" key while holding down the "CTRL" key. Special note: [1] is the numeral one — it is *not* the lower-case version of "L"!

Type the following sequence exactly to produce XFR\$:

[h] [Atari] [ ] [Atari] [CTRL-.] [Atari] [CTRL-E] [K] [CTRL-E] [M] [ ] [CTRL-.] [CTRL-E] [N] [%] [Atari] [j] [CTRL-X] [i] [CTRL-A] [Atari] [CTRL-E] [L] [spacebar] [Atari] [CTRL-.] [Atari] [1] [M] [CTRL-Q] [K] [H] [P] [y] [f] [L] [f] [N] [%] [N] [I] [d] [P] [m] [Atari] [CTRL-.]

## Variables

AZ: Miscellaneous uses.

A(\*): Equal to 1 if card already dealt in round.

BN\$: Pip values with black background.

CL: Time remaining on clock.

CLUB: Line number of routine that draws the club shape.

CP(\*): Seconds per card allowed for each player.

DL: Pointer to display list.

DIAMOND: Line number of routine that draws the diamond shape.

F1(\*): Value of card dealt, from ace=1 to king=13.

F2(\*): Suit of each card dealt, from 1 to 4.

G4: Number of games played.

HEART: Line number of routine that draws the heart shape.

HOME: Line number of the routine that clears the bottom of the screen.

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.

KBD: Address of the keyboard data location.

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.

NN\$: Pip values with normal background.

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: Used to GET input.

RN\$: Pip values with red background.

S(\*): Number of cards in each suit for hand. Example: S(1)=2 signifies that the hand contains two cards of suit one.

SCRN: Start of screen memory.

SPADE: Line number of routine that draws the spade shape.

START: Location of new character set.

T(\*): Number of cards by card value for hand. Example: T(5)=2 signifies that the hand contains a pair of fives.

U(\*): Number of a kind in hand. Example: U(2)=2 signifies that the hand contains two pairs.

V: Corresponding number of letter selected, at one point the same as the ASCII code, later adjusted (A=1, B=2, etc.).

X, X1, X2, X4, X5, X6, X7, X7, X8: For-loop variables.

XFR\$: Contains a Machine Language routine to rapidly download the ROM character set into RAM.

Y: Random number (1-52) for card dealt.

Z: For-loop variable.

Z1: Row number of letter selected on the board.

Z2: Column number of letter selected on the board.

Z3, Z8: For-loop variables.

Z9: Timing loop variable.

```

SS SS SS SS SS SS SS SS SS SS
SS                                     SS
SS      Atari BASIC                  SS
SS      'Poker Squares'              SS
SS      Authors: Ron O'Laughlin     SS
SS      Bob Whitworth                SS
SS      Translator: Alan J. Zett     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 also included on issue 41 SoftSide DV and CV.**

Initialization and introduction.

```

9 REM THIS PROGRAM CONTAINS TRUE LOWER
CASE CHARACTERS. IF YOU SEE NO REM BE-
FORE A LINE, TYPE IT VERBATIM.
10 GRAPHICS 0: DIM A(52), F1(25), F2(25),
L(5,5), M(5,5), P9(4), S(14), T(14), U(14),
CP(4), PB(4), P7(4): POKE 752, 1
20 OPEN #1, 4, 0, "K": POSITION 13, 5: ? "PO
KER SQUARES": POSITION 4, 8: ? "DO YOU WA
NT INSTRUCTIONS (Y/N)": HOME=140
30 SPADE=3000: CLUB=3010: DIAMOND=3020: H
EART=3030: DIM RN$(13), BN$(13), NN$(13):
GET #1, Q: IF Q=89 THEN GOSUB 1670
40 G4=0: FOR X=1 TO 4: P7(X)=0: NEXT X: KB
D=764: GOSUB 1790: NN$=CHR$(95): NN$(2)=C
HR$(34)
50 FOR X=3 TO 9: NN$(X)=CHR$(X+32): NEXT
X: FOR X=10 TO 13: NN$(X)=CHR$(X+81): NE
XT X: GOTO 250
60 DATA 72, 169, 0, 141, 10, 212, 141, 24, 208
, 169, 224, 141, 9, 212, 104, 64

```

Draw the game board.

```
110 POKE 559, 0: GOSUB 2680: RETURN
```

Zero out the card codes.

```
130 FOR X6=1 TO 14: S(X6)=0: T(X6)=0: U(X
6)=0: NEXT X6: RETURN
```

Erase text portion of screen.

```
140 AZ=USR(1664, SCRN+840, 120): RETURN
```

Shuffle the cards until the player hits a key.

```
150 GOSUB HOME: POSITION 7, 22: ? "SHUFFL
ING... PRESS ANY KEY": POKE KBD, 255

```

```
160 IF PEEK(KBD)=255 THEN Y=RND(0): GOTO
160
```

Zero out the scores and card-usage flag from previous games.

```
170 GOSUB HOME: FOR X=1 TO 4: P9(X)=0: PB
(X)=0: NEXT X
```

```
190 FOR X=1 TO 52: A(X)=0: NEXT X
```

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

```
200 FOR X=1 TO 25
```

```
210 Y=INT(RND(0)*52)+1
```

```
220 IF A(Y)=1 THEN 210
```

```
230 A(Y)=1: F2(X)=INT((Y-1)/13)+1: F1(X)
=Y-(F2(X)-1)*13
```

```
240 NEXT X: RETURN
```

Input the number of players.

```
250 GOSUB HOME: POKE 752, 1: POSITION 8, 2
2: ? "NUMBER OF PLAYERS? (1-4)":
```

```
260 GET #1, N: IF N<49 OR N>52 THEN 260
```

```
270 N=N-48: GOTO 320
```

```
280 ? CHR$(253): GOTO 330
```

Input each player's time limit.

```
320 FOR X=1 TO N: GOSUB HOME: POSITION 2
, 22: ? "PLAYER #": X: ? "SECONDS PER CARD
? (05-60)": CP(X)=0
```

```
330 GET #1, Q: IF Q<48 OR Q>57 THEN 280
```

```
340 GET #1, Y: IF Y<48 OR Y>57 THEN 280
```

```
350 Q=Q-48: Y=Y-48: IF Q*10+Y<5 OR Q*10+
Y>60 THEN 280
```

```
360 CP(X)=Q*10+Y: NEXT X: GOSUB HOME
```

Start play for each player.

```
410 GOSUB 150
```

```
420 FOR X=1 TO N: FOR X4=1 TO 5: FOR X5=
1 TO 5: L(X4, X5)=0: M(X4, X5)=0: NEXT X5: N
EXT X4: POKE KBD, 255
```

```
460 GOSUB HOME: POSITION 3, 22: ? "PLAYER
#": X: ? "PRESS ANY KEY TO START":
```

```
470 IF PEEK(KBD)=255 THEN 470
```

```
480 POKE KBD, 255
```

Draw the square for the "up" card.

```
490 FOR X6=8 TO 12: POSITION 29, X6: ? "
": NEXT X6
```

```
500 FOR X1=1 TO 25: GOSUB HOME: POSITION
29, 8: Q=F1(X1): ? NN$(Q, Q): POSITION 33
, 12: ? NN$(Q, Q):
```



## POKER SQUARES

```
510 B=30:Y=9:ON F2(X1) GOSUB SPADE,CLUB,DIAMOND,HEART
```

Check for card drop value.

```
530 POSITION 4,22: "PLACE YOUR CARD..  
  TIME LEFT: ";CP(X):CL=CP(X)
```

```
540 POKE 20,0
```

```
550 V=PEEK(KBD):IF V<255 THEN 660
```

```
560 IF PEEK(20)<60 THEN 550
```

Decrement the clock.

```
580 CL=CL-1:POSITION 34,22: CL,IF CL  
=3 THEN ? CHR$(253);
```

Beep when only three seconds are left.

```
610 IF CL>0 THEN 540
```

Assign penalty points when time runs out.

```
620 GOSUB HOME:POSITION 4,22: "OUT OF  
  TIME... 5 PENALTY POINTS!":PB(X)=PB(X)  
  )-5
```

```
630 FOR Z1=100 TO 70 STEP -0.7:SOUND 0  
  ,100-Z1,6,8:NEXT Z1:SOUND 0,0,0,0:FOR  
  Z1=0 TO 666:NEXT Z1:GOSUB HOME  
650 GOTO 530
```

Check for permissible card drop value.

```
660 IF PEEK(764)=154 THEN 560
```

```
670 GET #1,V
```

```
700 IF V<65 OR V>89 THEN 560
```

Adjust card drop value to the range 1-25, and convert it to a row and column, and check for card drop onto a previously used location.

```
740 V=V-64:Z1=INT((V-1)/5)+1:Z2=V-(Z1-  
  1)*5:IF L(Z1,Z2)=0 THEN 800
```

```
780 ? CHR$(253);:GOTO 560
```

Decode the card location.

```
800 L(Z1,Z2)=F1(X1):M(Z1,Z2)=F2(X1)
```

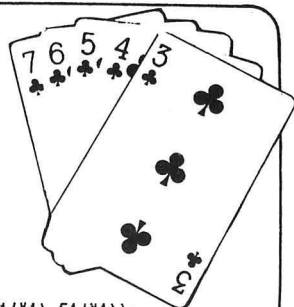
Remove the last "up" card.

```
810 FOR X6=8 TO 12:POSITION 29,X6: "  
  ":NEXT X6
```

Draw the card in the selected square.

```
830 Y=Z1*4-3:B=Z2*4-1:ON F2(X1) GOSUB  
  SPADE,CLUB,DIAMOND,HEART
```

```
840 POSITION B+1,Y+1:IF F2(X1)<3 THEN  
  ? BN*(F1(X1),F1(X1));:GOTO 910
```



```
850 ? RN*(F1(X1),F1(X1));
```

```
910 NEXT X1
```

Remove the "up" card square after the 25th card is placed.

```
920 FOR Y=8 TO 12:POSITION 28,Y: "idd  
  ddd!":NEXT Y:GOSUB HOME
```

Calculate score by row.

```
930 POSITION 2,21: "SCORES ACROSS: ";  
  :FOR X7=1 TO 5:GOSUB 130:FOR X8=1 TO 5  
  :S(M(X7,X8))=S(M(X7,X8))+1
```

```
970 T(L(X7,X8))=T(L(X7,X8))+1:NEXT X8:  
  GOSUB 1250:NEXT X7
```

```
1010 ? CHR$(30); "=";P:P9(X)=P9(X)+P
```

Calculate score by column.

```
1020 P=0:POSITION 2,22: "SCORES DOWN  
  : ";:FOR X7=1 TO 5:GOSUB 130:FOR X8=1  
  TO 5:S(M(X8,X7))=S(M(X8,X7))+1
```

```
1060 T(L(X8,X7))=T(L(X8,X7))+1:NEXT X8  
  :GOSUB 1250:NEXT X7
```

```
1100 ? CHR$(30); "=";P:P9(X)=P9(X)+P
```

Display penalty points and display final score.

```
1110 P=0: "PENALTY: ";PB(X);:POSITION  
  17,23:P9(X)=P9(X)+PB(X): "TOTAL SCORE  
  E: ";P9(X);:P7(X)=P7(X)+P9(X)
```

Wait for a keypress.

```
1130 GET #1,Q:IF X<>N THEN GOSUB 110  
1150 NEXT X
```

Display summary of scores.

```
1160 GRAPHICS 0:POSITION 10,4: "## S  
  U M M A R Y ##":G4=G4+1:POSITION 15,  
  6: " # GAMES = ";G4:?
```

```
1170 ? "PLAYER LAST ACCUM. A  
  VG.": "NUMBER GAME SCORE S  
  CORE"
```

```
1180 POKE 752,1: "=====
```



```
1190 FOR X=1 TO N: ? "NO. ";X; " ";P9
(X):POSITION 20,10+X: ? P7(X):POSITION
31,10+X: ? INT((P7(X)/64)+0.5):NEXT X
```

Play another hand?

```
1220 ? : ? : ? "PLAY ANOTHER HAND? (Y/N)
":GET #1,Q:IF Q<89 THEN POKE 106,PEEK
(106)+5:GRAPHICS 0:END
1230 GOSUB 110:GOTO 320
```

Determine number of each kind.

```
1250 FOR X5=1 TO 13
1260 IF T(X5)<2 THEN 1280
1270 U(T(X5))=U(T(X5))+1
1280 NEXT X5
```

Check for four of a kind.

```
1290 IF U(4)<1 THEN 1310
1300 ? "14+":P=P+16:RETURN
```

Check for a full house or three of a kind.

```
1310 IF U(3)<1 THEN 1350
1320 IF U(2)<1 THEN 1340
1330 ? "10+":P=P+10:RETURN
1340 ? "06+":P=P+6:RETURN
```

Check for number of pairs.

```
1350 IF U(2)=0 THEN 1390
1360 IF U(2)=2 THEN 1380
1370 ? "01+":P=P+1:RETURN
1380 ? "03+":P=P+3:RETURN
```

Check for a flush.

```
1390 K1=0
1400 FOR X5=1 TO 4
1410 IF S(X5)<5 THEN 1430
1420 K1=2
1430 NEXT X5
```

Check for a straight (ace high or low).

```
1440 K9=0:T(14)=T(1)
1450 FOR X5=1 TO 14
1460 IF K9=5 THEN 1530
1470 IF K9>0 THEN 1500
1480 IF T(X5)=0 THEN 1530
1490 K9=1:GOTO 1530
1500 IF T(X5)<>1 THEN 1520
1510 K9=K9+1:GOTO 1530
1520 K9=0
1530 NEXT X5
1540 IF K9=5 THEN K1=K1+1
```

Check for a "nothing" hand.

```
1550 IF K1>0 THEN 1570
1560 ? "00+":RETURN
```

Check for a straight flush.

```
1570 ON K1 GOTO 1580,1590,1600
1580 ? "12+":P=P+12:RETURN
1590 ? "05+":P=P+5:RETURN
1600 ? "30+":P=P+30:RETURN
```

Instructions.

```
1670 ? "}"INSTRUCTIONS":? : ? : ? "THE A
TARI DEALER WILL TURN UP 25 CARDS ONE A
T A TIME. YOU MUST SELECT ONE OF"
1680 ? "THE SQUARES LETTERED A-Y ON WH
ICH TO": ? "DROP EACH CARD."
```

```
1690 ? : ? "THE OBJECT IS TO CREATE 10
POKER HANDS": ? "OF FIVE CARDS EACH. T
HEY ARE LOCATED"
```

```
1700 ? "IN THE FIVE ROWS ACROSS AND AN
D THE": ? "FIVE COLUMNS DOWN."
```

```
1710 ? : ? "WHEN SEVERAL PLAYERS COMPET
E, ALL WILL BE DEALT THE SAME 25 CARDS.
WHERE YOU"
```

```
1720 ? "PUT THEM WILL DETERMINE THE SC
ORING": ? : ? : ? "HIT ANY KEY TO CONTIN
UE":GET #1,Q
```

```
1730 GRAPHICS 0:POKE 752,1: ? "INSTRUCT
IONS": ? : ? : ? "EACH HAND WILL EARN A
SCORE OF 0-30"
```

```
1740 ? "DEPENDING ON YOUR SKILL AND LU
CK": ? "GOOD PLAYERS MAY BE GIVEN A HA
NDICAP"
```

```
1750 ? "BY ALLOWING THEM LESS TIME TO
PLAY": ? "THE PENALTY IS FIVE POINTS E
VERY"
```

```
1760 ? "TIME THE TIMER REACHES ZERO":
? : ? : ? "AFTER EACH ROUND AN ACCUMULAT
ED SCORE"
```

```
1770 ? "SUMMARY WILL BE DISPLAYED FOR
EACH": ? "PLAYER TO SEE HOW WELL HE IS
DOING": ? : ?
```

```
1780 ? "GOOD LUCK AND NO CHEATING!": ?
: ? : ? "HIT ANY KEY TO CONTINUE":GET #
1,Q:RETURN
```

Routine to redefine the BASIC character set.

```
1790 POKE 106,PEEK(106)-5:GRAPHICS 0:P
OKE 559,0:START=(PEEK(106)+1)*256:POKE
756,START/256:POKE 752,1
```

## POKER SQUARES

```

1799 REM THE ARTICLE EXPLAINS HOW TO
TYPE XFR$
1800 DIM XFR$(38):XFR$=".....
....."
1810 Z=USR(ADR(XFR$)):RESTORE 1840
1820 READ X:IF X=-1 THEN 2660
1830 FOR Y=0 TO 7:READ Z:POKE X+Y*STAR
T,Z:NEXT Y:GOTO 1820

```

## Character definition data.

```

1840 DATA 16,0,42,2,42,32,34,42,0
1850 DATA 24,0,42,2,10,2,34,42,0
1860 DATA 32,0,34,34,42,2,2,2,0
1870 DATA 40,0,42,32,42,2,34,42,0
1880 DATA 48,0,42,32,42,34,34,42,0
1890 DATA 56,0,42,2,2,8,8,8,0
1900 DATA 64,0,42,34,42,34,34,42,0
1910 DATA 72,0,42,34,42,2,34,42,0
1920 DATA 144,255,171,251,171,191,187,
171,255
1930 DATA 152,255,171,251,235,251,187,
171,255
1940 DATA 160,255,187,187,171,251,251,
251,255
1950 DATA 168,255,171,191,171,251,187,
171,255
1960 DATA 176,255,171,191,171,187,187,
171,255
1970 DATA 184,255,171,251,251,239,239,
239,255
1980 DATA 192,255,171,187,171,187,187,
171,255
1990 DATA 200,255,171,187,171,251,187,
171,255
2000 DATA 264,255,239,187,187,171,187,
187,255
2010 DATA 272,255,175,187,175,187,187,
175,255
2020 DATA 280,255,239,187,191,191,187,
239,255
2030 DATA 288,255,175,187,187,187,187,
175,255
2040 DATA 296,255,171,191,175,191,191,
171,255
2050 DATA 304,255,171,191,175,191,191,
191,255
2060 DATA 312,255,235,191,191,187,187,
235,255

```

```

2070 DATA 320,255,187,187,171,187,187,
187,255
2080 DATA 328,255,171,239,239,239,239,
171,255
2090 DATA 336,255,251,251,251,187,187,
239,255
2100 DATA 344,255,187,187,175,175,187,
187,255
2110 DATA 352,255,191,191,191,191,191,
171,255
2120 DATA 360,255,187,171,171,187,187,
187,255
2130 DATA 368,255,175,187,187,187,187,
187,255
2140 DATA 376,255,239,187,187,187,187,
239,255
2150 DATA 384,255,175,187,187,175,191,
191,255
2160 DATA 392,255,239,187,187,187,175,
187,255
2170 DATA 400,255,175,187,187,175,187,
187,255
2180 DATA 408,255,171,191,171,251,251,
171,255
2190 DATA 416,255,171,239,239,239,239,
239,255
2200 DATA 424,255,187,187,187,187,187,
239,255
2210 DATA 432,255,187,187,187,187,239,
239,255
2220 DATA 440,255,187,187,187,171,171,
187,255
2230 DATA 448,255,187,187,239,239,187,
187,255
2240 DATA 456,255,187,187,187,239,239,
239,255
2250 DATA 472,0,42,8,8,8,8,8,0
2260 DATA 480,0,2,2,2,34,34,8,0
2270 DATA 488,0,8,34,34,34,40,34,0
2280 DATA 496,0,34,34,40,40,34,34,0
2290 DATA 504,0,8,34,34,42,34,34,0
2300 DATA 520,247,247,247,245,245,247,
247,247
2310 DATA 536,247,247,247,87,87,255,25
5,255
2320 DATA 544,247,247,247,87,87,247,24
7,247
2330 DATA 552,255,255,255,87,87,247,24
7,247

```

```

2340 DATA 648,255,255,255,245,245,247,
247,247
2350 DATA 656,255,255,255,85,85,255,25
5,255
2360 DATA 664,247,247,247,85,85,247,24
7,247
2370 DATA 696,255,255,255,85,85,247,24
7,247
2380 DATA 704,247,247,247,85,85,255,25
5,255
2390 DATA 720,247,247,247,245,245,255,
255,255
2400 DATA 776,0,0,0,0,0,3,3,3
2410 DATA 784,0,48,48,252,252,255,255,
255
2420 DATA 792,15,15,15,63,63,63,63,63
2430 DATA 800,255,255,255,255,255,255,
255,255
2440 DATA 808,192,192,192,240,240,240,
240,240
2450 DATA 816,63,63,15,0,0,3,15,0
2460 DATA 824,255,255,51,48,252,255,25
5,0
2470 DATA 832,240,240,192,0,0,0,192,0
2480 DATA 840,0,0,0,3,3,3,3,3
2490 DATA 848,0,0,252,255,255,255,255,
255
2500 DATA 856,3,0,15,63,63,63,63,63
2510 DATA 864,255,252,51,255,255,255,2
55,255
2520 DATA 872,0,0,192,240,240,240,240,
240
2530 DATA 880,0,0,0,3,15,63,63,63
2540 DATA 888,0,0,0,3,207,255,255,255
2550 DATA 896,0,0,0,0,192,240,240,240
2560 DATA 904,63,63,63,63,63,15,15,3
2570 DATA 912,240,240,240,240,240,192,
192,0
2580 DATA 920,3,0,0,0,0,0,0,0
2590 DATA 928,255,252,252,48,48,0,0,0
2600 DATA 936,0,0,0,0,0,0,0,3
2610 DATA 944,0,0,0,48,48,252,252,255
2620 DATA 952,3,15,15,63,63,15,15,3
2630 DATA 960,0,192,192,240,240,192,19
2,0
2640 DATA 992,247,247,247,247,247,247,
247,247
2650 DATA -1

```

## Redefine display list.

```

2660 DL=PEEK(560)+PEEK(561)*256+4:REST
ORE 2700:FOR X=1664 TO 1741:READ Y:POK
E X,Y:NEXT X:BN$="A23456789TJQK"

```

Poke in machine language routine to zero memory.

```

2670 RESTORE 60:FOR X=1750 TO 1765:REA
D Y:POKE X,Y:NEXT X
2680 POKE 756,START/256:POKE DL-1,68:F
OR X5=2 TO 20:POKE DL+X5,4:NEXT X5:POK
E DL+21,132

```

Set up background colors.

```

2690 POKE 712,8:POKE 711,66:POKE 710,0
:POKE 709,156:POKE 708,146:POKE 512,21
4:POKE 513,6:POKE 54286,192

```

Data for machine language zero routine.

```

2700 DATA 160,0,104,141,205,6,104,133,
204,104,133,203,104,141,204,6,104,141,
203,6,173,205,6,201,2,208
2710 DATA 7,169,0,141,205,6,240,5,104,
104,141,205,6,173,203,6,13,204,6,240,2
7,173,205,6,145,203
2720 DATA 230,203,208,2,230,204,206,20
3,6,173,203,6,201,255,208,227,206,204,
6,24,144,221,96,0,0,0

```

Set up original game board.

```

2730 SCRN=PEEK(88)+PEEK(89)*256:AZ=USR
(1664,SCRN,840,100):COLOR 228:FOR Y=1
TO 19:PLOT 25,Y:DRAWTO 37,Y:NEXT Y
2739 REM NEXT LINE HAS CTRL CHARS, BUT
THE d CHARACTERS ARE TRUE LOWER CASE.
2740 PLOT 2,0:POSITION 2,0:?"qrrrrrrr
rrrrrrrrrrrr":FOR X5=1 TO 5:FOR Y=1 T
O 3:?"iddd:ddd:ddd:ddd:ddd":NEXT Y
2749 REM NEXT 2 LINES CONTAIN CONTROL
CHARACTERS. ALL LOWER CASE MUST BE
TYPED AS CTRL.
2750 ?"arrrrrrrrrrrrrrrrrrrrrr":NEXT X5
:?"CHR$(28):"zrrrrrrrrrrrrrrrrrrrrrr"
2760 POSITION 29,2:?"POKER":POSITION
28,3:?"SQUARES":POSITION 28,7:?"qrrr
rrr":POSITION 28,13:?"zrrrrrrr"
2770 FOR Y=8 TO 12:POSITION 28,Y:?"id
ddd":NEXT Y:RN$="A23456789TJQK":GOSU
B HOME

```

## POKER SQUARES

```
2780 FOR Y=0 TO 4:FOR X5=0 TO 4:POSITI
ON X5*4+4,Y*4+2:? CHR$(Y*5+X5+65);:NEX
T X5:NEXT Y:POKE 559,34:RETURN
```

Subroutines for drawing card shapes from redefined characters.

```
3000 POSITION Q,Y:? "ab ":POSITION Q,Y
+1:? "cde":POSITION Q,Y+2:? "fgh":RETU
RN
```

```
3010 POSITION Q,Y:? "ij ":POSITION Q,Y
+1:? "klm":POSITION Q,Y+2:? "fgh":RETU
RN
```

```
3020 POSITION Q,Y:? "uv ":POSITION Q,Y
+1:? "wdx":POSITION Q,Y+2:? "st ":RETU
RN
```

```
3030 POSITION Q,Y:? "nop":POSITION Q,Y
+1:? "qdr":POSITION Q,Y+2:? "st ":RETU
RN
```

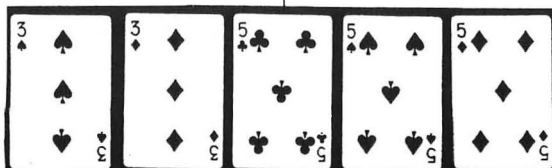


# SWAT TABLE

For ATARI® POKER SQUARES

LINES	SWAT CODE	LENGTH
9 - 20	JU	423
30 - 50	WE	337
60 - 150	XA	277
160 - 210	QS	206
220 - 260	BN	243
270 - 340	ES	246
350 - 460	WM	303
470 - 510	CD	224
530 - 580	BJ	221
610 - 660	ZF	284
670 - 800	IJ	210
810 - 910	RE	203
920 - 1020	QL	398
1060 - 1150	EI	237
1160 - 1190	ER	367
1220 - 1270	AC	206
1280 - 1320	BH	132
1330 - 1370	BY	151
1380 - 1420	GK	108
1430 - 1470	LH	115
1480 - 1520	KD	119
1530 - 1570	WY	105
1580 - 1680	RY	259
1690 - 1720	PF	335

LINES	SWAT CODE	LENGTH
1730 - 1760	CR	341
1770 - 1790	RG	304
1799 - 1830	QE	209
1840 - 1880	FA	143
1890 - 1930	MF	166
1940 - 1980	AJ	205
1990 - 2030	MN	205
2040 - 2080	GY	205
2090 - 2130	IK	205
2140 - 2180	SK	205
2190 - 2230	CZ	205
2240 - 2280	KB	155
2290 - 2330	II	188
2340 - 2380	SH	197
2390 - 2430	JA	177
2440 - 2480	QW	165
2490 - 2530	UZ	174
2540 - 2580	AG	162
2590 - 2630	LM	159
2640 - 2680	SI	306
2690 - 2710	JA	335
2720 - 2740	FF	428
2749 - 2770	VI	353
2780 - 3020	OK	328
3030 - 3030	FH	64



# Five Card Stud Poker



by Richard Spencer

**Five Card Stud Poker is a card game for an Atari® 400/800/1200 with 24K (tape), 32K (disk) and DOS II. It is included as the bonus program on Issue 41 Atari DV. See the coupon near the back of this booklet to order your disk.**

Are your nerves frazzled by noisy, demanding arcade shoot-em-ups? Try the excitement of betting big bucks on the turn of a single card. Try waiting a tense moment to see if your big raise on a pair of queens will bluff a pair of kings out of the game. The program plays five card stud with exact mathematical duplication of the real game. You play against four computer-controlled players whose different personalities range from reckless, to wimp, to cool, calculating professional.

Your input is easy. Whenever you see a flashing happy face, use your joystick to position the face under your instruction, and press the red button. When the game starts, type your name, and the amount of money you need to start. Since the object of the game is to win all the money from all the players, staying under \$500 is a good idea.

Each player antes five dollars. The dealer deals a down card and an up card to each player. The down card is known only to the player who holds it. Betting starts with the highest showing card. His options are to bet either five dollars or ten dollars, or to fold. If he folds, the next player has the option of first bet. If he bets, each player in turn can opt to call (bet the same amount), raise or fold. The game allows only three raises. After the bets are equalized, another up card is dealt, followed by another betting interval and so on until each player has five cards. If all but one of the players fold, the remaining player wins and need not show his down card. However, if two or more players stay in the game until the fourth betting interval is finished, a showdown is held. Each player exposes his down card. The highest ranking hand wins.

**DV BONUS**

## FIVE CARD STUD POKER

DV BONUS



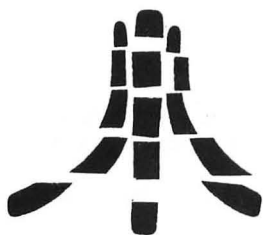
The hands rank from lowest to highest as follows:

- **High Card:** The highest card in your hand. You have no matched cards, no cards in an unbroken sequence, and the suits are mixed. The computer will deal you this hand about half the time.
- **One Pair:** A pair of matched cards. The higher pair, such as kings, beats a lower pair, such as tens. On the average you will get a pair a little less than half the time.
- **Two Pair:** Two pairs of matched cards. Occurs once in twenty hands.
- **Three of a Kind:** Occurs about once in 46 hands.
- **Straight:** Five cards in sequences. Occurs once in 254 hands.
- **Flush:** Five unmatched cards all of the same suit, such as hearts. Occurs once in 508 hands.
- **Full House:** Three of a kind and a pair. Occurs once in 693 hands.
- **Four of a Kind:** Not very frequent — on the average of once in 4,164 hands.
- **Straight Flush:** Five cards in sequence and of the same suit. Seen once in about 72,192 hands.

---

### Strategy

Bet high when your hand looks like the winner. Call when you have a lesser chance. Fold when you are beaten by a showing hand. Players with strong hands will usually raise. If you have a weak hand, and two or more players raise, that's usually a good time to fold. However, you will not always know when the other players have the strong hand. They will, on occasion, bluff. You can bluff also, but make sure your hand looks good. Don't try to bluff a pair of aces with a ten high hand. It is also ethical to bet low with a sure hand, to avoid frightening away the money.



# Atari

# Blaster

by Karl Haferer

**Atari Blaster is a one-player arcade-style game for an Atari 400/800/1200 computer with 16K RAM and a joystick.**

*Atari Blaster* pits you against nasty aliens intent on pilfering your most prized possession, your computer. The enemy is attempting to dock with your space station in order to steal your machine. They can approach your base from only four directions: the top, bottom, left and right sides of your screen.

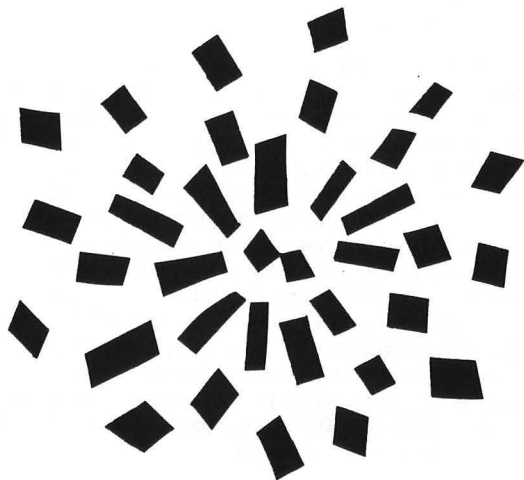
You control your weapon, the blaster, with a joystick plugged into port one. To move the blaster, push the stick in the desired direction. To fire, press the fire button. In addition to your normal blaster, you have one super blast at the start of every game. It destroys all of the enemy vessels on the screen when you press any key on the keyboard. Every 1500 points, you earn an additional super blast, and you can accumulate up to three.

You get ten, fifteen, or twenty points when you destroy one of the enemy ships. The high score appears at the top left of the screen, with your current game score displayed underneath. The other side of the screen displays the number of remaining super blasts. If you want to skip the low levels of the game, press **OPTION**. The enemy ships then increase their speed. If you wish to restart the game at any time, press **START**. To pause the game, press **CTRL-1**. To resume play, press the key again.

*Note:* If you play long enough without using a super blast, your Atari goes into its screen color changing routine. To remedy this, press the Atari Logo key (A).



## ATARI BLASTER



## Variables

**A6:** Constant 6, used to enhance execution speed.

**DE:** Direction Atari is being stolen.

**DEL:** Delay loop.

**HSC:** The high score.

**I, L:** Loops.

**MO:** The movement value added to the enemy ship's position.

**P1:** Value of ship No.1's Y position.

**P2:** Value of ship No.2's X position.

**P3:** Value of ship No.3's Y position.

**P4:** Value of ship No.4's X position.

**PO:** The position to put the blaster when it is to be moved.

**S1:** Value to be added to the score if the ship in direction 1 is hit.

**S2:** Value to be added to the score if the ship in direction 2 is hit.

**S3:** Value to be added to the score if the ship in direction 3 is hit.

**S4:** Value to be added to the score if the ship in direction 4 is hit.

**SB:** The number of remaining super blasts.

**SC:** Your current game score.

**SH(\*):** In this array, the value can be 1 or 0. A 1 indicates that a ship is approaching in the direction (\*); 0 means there is no ship.

**SL:** The sum of all of the SH array values. If it is greater than 0, there are still ships on the screen to move.

**STI:** Value of the joystick when it is read.

**STR:** A variable used as a marker for redrawing the blaster.

**SZ:** Variable to keep track of your score to see if you obtain a free super blast.

**X:** Controls loops, and obtains a keyboard PEEK.

**Y:** Used within loops to read DATA.

```

SS SS SS SS SS SS SS SS SS SS
SS                                     SS
SS      Atari BASIC                  SS
SS      'Atari Blaster'              SS
SS      Author: Karl Haferer        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.

Delay loop.

```

10 DIM SH(4):GOSUB 625:A6=6:HSC=0
15 GOSUB 605:M0=0:SC=0:SB=1:SZ=0:POKE
764,255:GOTO 230
20 FOR DEL=1 TO 20:NEXT DEL:RETURN

```

Laser sound.

```

25 FOR I=8 TO 1 STEP -1:SOUND 0,I*5,10
,12:NEXT I:SOUND 0,0,0,0:RETURN

```

Move the ships and check for docking.

```

35 POSITION 9,P1:? #A6;" ":P1=P1+M0:I
F P1>10 THEN P1=10:POP :DE=1:GOTO 505
40 POSITION 9,P1:? #A6;"AB":RETURN
45 POSITION P2,11:? #A6;" ":POSITION P
2,12:? #A6;" ":P2=P2-M0:IF P2<11 THEN
P2=11:POP :DE=2:GOTO 505
50 POSITION P2,11:? #A6;"G":POSITION P
2,12:? #A6;"H":RETURN
55 POSITION 9,P3:? #A6;" ":P3=P3-M0:I
F P3<13 THEN P3=13:POP :DE=3:GOTO 505
60 POSITION 9,P3:? #A6;"DC":RETURN
65 POSITION P4,11:? #A6;" ":POSITION P
4,12:? #A6;" ":P4=P4+M0:IF P4>8 THEN P
4=8:POP :DE=4:GOTO 505
70 POSITION P4,11:? #A6;"E":POSITION P
4,12:? #A6;"F":RETURN

```

Fire and see if a ship is destroyed.

```

80 POSITION 9,10:? #A6;"i":FOR I=9 TO
INT(P1+0.5)+1 STEP -1:POSITION 9,I:? #
A6;"o":NEXT I:GOSUB 25
85 IF SH(1) THEN POSITION 9,P1:? #A6;"
##":SC=SC+S1:SZ=SZ+S1

```

```

90 GOSUB 190:FOR I=9 TO INT(P1+0.5)+1
STEP -1:POSITION 9,I:? #A6;" ":NEXT I
95 IF SH(1) THEN POSITION 9,P1:? #A6;"
%&":GOSUB 20:POSITION 9,P1:? #A6;" ":
SH(1)=0:P1=0:GOSUB 190
100 RETURN
105 POSITION 11,11:? #A6;"u":FOR I=12
TO INT(P2-0.5):? #A6;"v":NEXT I:GOSU
B 25
110 IF SH(2) THEN POSITION P2,11:? #A6
;" ":POSITION P2,12:? #A6;"(":SC=SC+S2
:SZ=SZ+S2
115 GOSUB 200:FOR I=12 TO INT(P2-0.5):
POSITION I,11:? #A6;" ":NEXT I:IF NO
T SH(2) THEN RETURN
120 POSITION P2,11:? #A6;" ":POSITION
P2,12:? #A6;"%&":GOSUB 20:POSITION P2,1
1:? #A6;" ":POSITION P2,12:? #A6;" "
125 SH(2)=0:P2=19:GOSUB 200:RETURN
130 POSITION 9,13:? #A6;"n":FOR I=14 T
O INT(P3-0.5):POSITION 9,I:? #A6;"o":N
EXT I:GOSUB 25
135 IF SH(3) THEN POSITION 9,P3:? #A6;
"##":SC=SC+S3:SZ=SZ+S3
140 GOSUB 210:FOR I=14 TO INT(P3-0.5):
POSITION 9,I:? #A6;" ":NEXT I
145 IF SH(3) THEN POSITION 9,P3:? #A6;
"%&":GOSUB 20:POSITION 9,P3:? #A6;" "
:SH(3)=0:P3=23:GOSUB 210
150 RETURN
155 POSITION 8,11:? #A6;"r":FOR I=7 TO
INT(P4+0.5)+1 STEP -1:POSITION I,11:?
#A6;"v":NEXT I:GOSUB 25
160 IF SH(4) THEN POSITION P4,11:? #A6
;" ":POSITION P4,12:? #A6;"(":SC=SC+S4
:SZ=SZ+S4
165 GOSUB 220:FOR I=7 TO INT(P4+0.5)+1
STEP -1:POSITION I,11:? #A6;" ":NEXT
I:IF NOT SH(4) THEN RETURN
170 POSITION P4,11:? #A6;"j":POSITION
P4,12:? #A6;"%&":GOSUB 20:POSITION P4,1
1:? #A6;" ":POSITION P4,12:? #A6;" "
175 SH(4)=0:P4=0:GOSUB 220:RETURN

```

Move the blaster according to input.

```

185 POSITION 9,10:? #A6;" ":RETURN
190 POSITION 9,10:? #A6;"jk":RETURN

```

## ATARI BLASTER

## ATARI BLASTER

```

195 POSITION 11,11: ? #A6; " ":POSITION
11,12: ? #A6; " ":RETURN
200 POSITION 11,11: ? #A6; "5":POSITION
11,12: ? #A6; "t":RETURN
205 POSITION 9,13: ? #A6; " ":RETURN
210 POSITION 9,13: ? #A6; "m1":RETURN
215 POSITION 8,11: ? #A6; " ":POSITION 8
,12: ? #A6; " ":RETURN
220 POSITION 8,11: ? #A6; "p":POSITION 8
,12: ? #A6; "q":RETURN

```

Main loop from which all subroutines are accessed.

Determine the speed of the enemy vessels. This can be changed to your personal liking.

```

230 P1=0:P2=19:P3=23:P4=0:GOSUB 370:IF
  NO<1.4 THEN NO=NO+0.12:GOTO 240
235 NO=NO+0.02

```

Init loop.

```

240 FOR L=1 TO 4

```

If SH(\*)=1 then a ship is present. GOSUB to move it.

```

245 IF SH(L) THEN ON L GOSUB 35,45,55,
65

```

Get user input.

Read joystick to determine blaster movement and move it.

```

250 STI=STICK(0):IF STI=15 OR STI=10 OR
  STI=6 OR STI=5 OR STI=9 THEN 280
255 IF STI=14 THEN PO=1:GOTO 275
260 IF STI=7 THEN PO=2:GOTO 275
265 IF STI=13 THEN PO=3:GOTO 275
270 IF STI=11 THEN PO=4
275 IF STR<>PO THEN ON STR GOSUB 185,1
95,205,215:ON PO GOSUB 190,200,210,220
:STR=PO

```

Checks button for fire.

```

280 IF STRIB(0)=0 THEN ON PO GOSUB 80,
105,130,155

```

Checks inverse video key.

```

285 X=PEEK(764):IF X=39 THEN POKE 764,
255:GOTO 295

```

If any other key has been pressed, award one super blast.

```

290 IF X<255 AND SB>0 THEN GOSUB 330:G
OTO 275

```

Checks for a bonus super blast.

```

295 IF SZ>1499 AND SB<3 THEN SZ=SZ-150
0:SOUND 0,75,10,8:GOSUB 20:GOSUB 20:SO
UND 0,0,0,0:SB=SB+1

```

If number of super blasts is three, then award no more free ones.

```

300 IF SB=3 THEN SZ=0

```

Checks START key.

```

305 IF PEEK(53279)=6 THEN POP :GOTO 15

```

Checks OPTION key.

```

307 IF PEEK(53279)=3 AND MO<2 THEN SOU
ND 0,30,12,15:FOR X=1 TO 100:NEXT X:SO
UND 0,0,0,0:MO=2

```

Print out the scores.

```

310 GOSUB 340:POSITION 5-LEN(STR$(SC))
,2: ? #A6;SC:IF SC>HSC THEN HSC=SC:POSI
TION 7-LEN(STR$(SC)),0: ? #A6;SC
315 NEXT L:FOR L=1 TO 4:SL=SL+SH(L):NE
XT L:IF SL>0 THEN SL=0:GOTO 240
320 SL=0:GOTO 230

```

Super blast routine.

```

330 L=4:SOUND 1,46,12,10:ON PO GOSUB 1
85,195,205,215:FOR X=1 TO 4:IF SH(X) T
HEN ON X GOSUB 345,350,355,360
335 NEXT X:STR=0:SOUND 1,0,0,0:SB=SB-1
340 POSITION 17,0: ? #A6;SB:POKE 764,25
5:RETURN
345 GOSUB 190:GOSUB 80:GOSUB 185:RETUR
N
350 GOSUB 200:GOSUB 105:GOSUB 195:RETU
RN
355 GOSUB 210:GOSUB 130:GOSUB 205:RETU
RN
360 GOSUB 220:GOSUB 155:GOSUB 215:RETU
RN

```

Choose the attacking ships.

```

370 IF RND(0)>0.75 THEN SH(1)=0:GOTO 3
80
375 RESTORE INT(RND(0)*5+1)*5+420:FOR
X=0 TO 15:READ Y:POKE ST+264+X,Y:NEXT
X:SH(1)=1:READ S1
380 IF RND(0)>0.75 THEN SH(2)=0:GOTO 3
90
385 RESTORE INT(RND(0)*5+1)*5+445:FOR
X=0 TO 7:READ Y:POKE ST+312+X,Y:POKE S
T+327-X,Y:NEXT X:SH(2)=1:READ S2

```

```

390 IF RND(0)>.75 THEN SH(3)=0:GOTO 4
00
395 RESTORE INT(RND(0)*5+1)*5+420:FOR
X=0 TO 15:READ Y:POKE ST+295-X,Y:NEXT
X:SH(3)=1:READ S3
400 IF RND(0)>.75 THEN SH(4)=0:GOTO 4
10
405 RESTORE INT(RND(0)*5+1)*5+470:FOR
X=0 TO 7:READ Y:POKE ST+296+X,Y:POKE S
T+311-X,Y:NEXT X:SH(4)=1:READ S4
410 FOR L=1 TO 4:IF SH(L) THEN ON L GO
SUB 40,50,60,70
415 NEXT L:RETURN
Data for the shapes.
425 DATA 25,31,27,30,30,23,17,49,152,2
48,216,120,120,232,136,140,10
430 DATA 31,23,19,25,49,7,4,6,248,232,
200,152,140,224,32,96,10
435 DATA 40,63,47,44,38,35,33,1,20,252
,244,52,100,196,132,128,15
440 DATA 4,31,127,87,87,83,83,65,64,24
0,252,212,212,148,148,4,10
445 DATA 71,47,41,57,57,41,47,6,226,24
4,148,156,156,148,244,96,20
450 DATA 0,128,255,15,31,58,62,231,10
455 DATA 0,16,31,9,3,227,167,63,10
460 DATA 0,127,2,6,15,26,50,226,15
465 DATA 0,252,4,126,6,31,126,254,10
470 DATA 1,126,24,24,126,195,195,127,2
0
475 DATA 0,1,255,240,248,92,124,231,10
480 DATA 0,8,248,144,192,199,229,252,1
0
485 DATA 0,254,64,96,240,88,76,71,15,1
5
490 DATA 0,63,32,126,96,248,126,127,10
495 DATA 128,126,24,24,126,195,195,254
,20
Steal the Atari and destroy space
station.
505 MD=-1:ON PD GOSUB 185,195,205,215:
ON DE GOSUB 35,45,55,65:FOR L=1 TO 254
STEP 2:SOUND 0,L,12,10:NEXT L
510 POSITION 9,11: ? #A6;"WX":POSITION
9,12: ? #A6;"YZ":ON DE GOTO 515,525,535
,545

```

```

515 POSITION 9,P1: ? #A6;">":FOR L=1 T
O 9:GOSUB 35:POSITION 9,P1+2: ? #A6;"
":POSITION 9,P1+1: ? #A6;">?"
520 GOSUB 20:NEXT L:POSITION 9,P1+1: ?
#A6;" ":POSITION 11,0: ? #A6;">":A=1:
B=10:C=1:GOTO 565
525 GOSUB 45:POSITION P2-2,11: ? #A6;">
?":FOR L=1 TO 6:GOSUB 45:POSITION P2-3
,11: ? #A6;">?"
530 GOSUB 20:NEXT L:POSITION P2-2,11: ?
#A6;" ":POSITION 18,10: ? #A6;">":A=
18:B=11:C=-1:GOTO 555
535 POSITION 9,P3: ? #A6;"DC":FOR L=1 T
O 9:GOSUB 55:POSITION 9,P3-2: ? #A6;"
":POSITION 9,P3-1: ? #A6;">?"
540 GOSUB 20:NEXT L:POSITION 9,P3-1: ?
#A6;" ":POSITION 11,23: ? #A6;">":A=2
2:B=13:C=-1:GOSUB 565
545 GOSUB 65:POSITION P4+1,11: ? #A6;">
?":FOR L=1 TO 6:GOSUB 65:POSITION P4+1
,11: ? #A6;">?"
550 GOSUB 20:NEXT L:POSITION P4+1,11: ?
#A6;" ":POSITION 0,10: ? #A6;">":A=1
:B=8:C=1
555 FOR L=A TO B STEP C:POSITION L,11:
? #A6;"V":NEXT L:GOSUB 25:FOR L=A TO B
STEP C:POSITION L,11: ? #A6;" "
560 NEXT L:GOTO 570
565 FOR L=A TO B STEP C:POSITION 9,L: ?
#A6;"O":NEXT L:GOSUB 25:FOR L=A TO B
STEP C:POSITION 9,L: ? #A6;" ":NEXT L
570 GOSUB 20:POSITION 9,11: ? #A6;"##":
POSITION 9,12: ? #A6;"##":GOSUB 20:POSI
TION 9,11: ? #A6;"%&":POSITION 9,12
575 ? #A6;"%&":GOSUB 20:POSITION 9,11:
? #A6;"'":POSITION 9,12: ? #A6;"('":GO
SUB 20:POSITION 9,11: ? #A6;"')")
580 POSITION 9,12: ? #A6;"##":GOSUB 20:
POSITION 9,11: ? #A6;" ":POSITION 9,12
: ? #A6;" "
585 FOR L=1 TO 250:NEXT L:POSITION 8,1
1: ? #A6;"-.[ ":POSITION 8,12: ? #A6;"\
j^_ "
590 IF PEEK(53279)<>6 AND STRIG(0)=1 T
HEN 590
595 GOTO 15

```

## ATARI BLASTER

Initialize graphics screen, draw stars, etc.

```
605 GRAPHICS 17:POKE 756,ST/256:FOR X=
1 TO 100:POSITION RND(0)*18,RND(0)*22:
? #6;"!":NEXT X
610 POSITION 2,0:? #6;"00000":? #6:? #
6;"00000":POSITION 9,11:? #6;"":POS
ITION 9,12:? #6;"<="
615 POSITION 7-LEN(STR$(HSC)),0:? #A6;
HSC:PO=INT(RND(0)*4+1):STR=PO:ON STR G
OSUB 190,200,210,220:RETURN
```

Print out the title screen, copy internal character set and POKE in new character shapes.

```
625 POKE 106,PEEK(106)-5:ST=(PEEK(106)
+1)*256
630 GRAPHICS 17:POSITION 3,6:? #6;"ATA
RI BLASTER":POSITION 8,8:? #6;"BY":POS
ITION 3,10:? #6;"karl haferer"
635 POSITION 3,13:? #6;"PLEASE WAIT..."
FOR X=0 TO 1023:POKE ST+X,PEEK(57344
+X):NEXT X:RESTORE 670
640 FOR X=0 TO 63:READ Y:POKE ST+24+X,
Y:NEXT X:FOR X=0 TO 47:READ Y:POKE ST+
208+X,Y:NEXT X
645 FOR X=0 TO 23:READ Y:POKE ST+328+X
,Y:POKE ST+375-X,Y:NEXT X:FOR X=0 TO 1
35:READ Y:POKE ST+376+X,Y:NEXT X
650 FOR X=0 TO 7:READ Y:POKE ST+8+X,Y:
NEXT X:FOR X=0 TO 15:READ Y:POKE ST+10
4+X,Y:NEXT X
655 POSITION 3,13:? #6;"press trigger"
"
660 IF PEEK(53279)<>6 AND STRIG(0)=1 T
HEN 660
665 RETURN
```

The DATA for redefinition.

```
670 DATA 0,0,5,0,2,4,0,0,0,160,0,192,1
6,128,0,0,0,36,17,12,42,1,36,16,8,160,
32,140,96,20,0,64
675 DATA 0,0,0,0,0,16,68,16,10,32,68,1
6,0,0,0,0,33,66,17,74,36,136,2,40,37,8
2,4,74,24,0,69,16
680 DATA 31,63,96,192,197,197,197,197,
248,252,6,3,163,163,163,163,197,197,20
1,209,192,96,63,31,163,163,147,139
685 DATA 3,6,252,248,5,5,5,5,5,9,17,
160,160,160,160,160,160,144,136
```

```
690 DATA 1,1,1,1,3,7,15,31,0,0,0,0,3,7
,15,31,0,0,0,0,192,224,240,248,1,1,1,1
,1,1,1,1,0,0,0,1,3,7,15,15,15,15,7,3
695 DATA 1,0,0,0,0,0,0,1,3,7,15,255,0,
0,0,128,192,224,240,240,240,240,224,19
2,128,0,0,0,0,0,0,128,192,224,240,255
700 DATA 0,0,0,0,0,0,0,255,31,63,96,19
2,192,192,192,192,248,252,6,3,3,3,3,3,
192,192,192,192,192,96,63,31
705 DATA 3,3,3,3,3,6,252,248,0,0,224,1
28,64,32,224,0,0,29,9,9,9,9,0,0,156,
73,137,73,92,0,0,0,198,8,107,41,198,0
710 DATA 0,0,118,69,102,69,117,0,0,0,0
,0,0,8,0,0,0,0,204,170,204,138,138,0
,0,0,238,136,196,130,238,0
```



For ATARI® ATARI BLASTER

LINES	SWAT CODE	LENGTH
10 - 50	WI	538
55 - 90	SZ	501
95 - 130	OW	590
135 - 165	SD	514
170 - 220	EG	522
230 - 280	MR	546
285 - 310	CY	537
315 - 360	HR	509
370 - 395	KK	541
400 - 440	YB	512
445 - 505	QI	555
510 - 530	XF	557
535 - 555	BO	531
560 - 585	TZ	527
590 - 630	UA	630
635 - 670	FE	588
675 - 700	DY	583
705 - 710	VT	209



# 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™ 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 \_\_\_\_\_

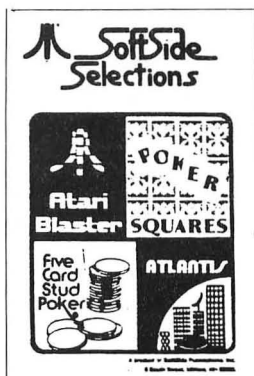
☐ Check or Money Order

☐ MasterCard    ☐ VISA

Name of Cardholder \_\_\_\_\_

MC# or Interbank#/VISA# \_\_\_\_\_

Exp. Date \_\_\_\_\_ Signature \_\_\_\_\_



Prices subject to change without notice Apple®, IBM®, Atari® and TRS-80® are registered trademarks of The Apple Computer Company, International Business Machines, Inc., Warner Communications, and The Tandy Corporation respectively.

# SoftSide<sup>TM</sup> 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.

In 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 Atari with at least 32K RAM (40K disk).**

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



MZGREVH ORPV ZOXL SLO.

IFYYVI RH Z TLLW RMHFOZGLI.

DSZG WL ZM LXGLKFH ZMW Z GBKVDIRGVI  
IRYYLM SZE V RM XLNNLM?

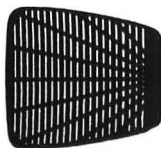
GSV MZGREVH ZIV ZUIZRW LU GSV XZEV.

TIZHH ZMW YZNYLL NZPV Z TLLW GIZK.

HVZTFOO UVZGSVIH NZPV TLLW KVMH.

## General Information About Listings, SWAT, and Magnetic Media

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 Magazine** may differ in some details.



## SWAT TABLE

### SWAT Tables

At the conclusion of each program listing in **SoftSide Selections**, we include a **SWAT (Strategic Weapon Against Typos)** Table. **SWAT** for the Atari 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

Be sure to tell us that you have an Atari computer.

### Magnetic Media

Disks do not carry the **DOS.SYS** and **DUP.SYS** files, and are not "bootable." First, boot a disk with **DOS** on it, then insert the **SoftSide Selections** disk, and run "**D:COVER**". Our disks are in **DOS 2** format.

Tapes **CLOAD** in the normal manner. If you encounter difficulty, try this procedure:

1. **POKE 54018,54**
2. Turn up the volume on your TV.
3. Type **CLOAD**, and press **RETURN** once.
4. Press the play button, and listen.
5. When you hear the steady leader tone, press **RETURN** again.

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.

**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

Line listings are in standard 38-column format, with special conventions for representing unprintable characters:

You must type underlined characters, including blank spaces, in inverse video.

When graphics or control (CTRL) characters are included in a string (between quotation marks), a nearby REM statement will make note of it; in such cases, graphics characters appear as the corresponding lower-case letters, and control characters appear as the corresponding unshifted key symbols. For example: the lower-case letter **s** represents a graphic cross, which you type by pressing the S key while holding down the CTRL key; the **=** sign represents CTRL-down-arrow, which you type by pressing and releasing the ESC key, then pressing the **=** key while holding down CTRL. For more information about entering control characters, refer to Appendix F and the back cover of your **Atari BASIC Reference Manual**.

There are two exceptions to our above convention: A clear-screen character (ESC SHIFT-CLEAR) appears in our listings as a right-hand brace, which looks like this: **}**. The other exception is that a shifted **=** sign appears as a broken vertical line: **|**.

Occasionally, a program will demand that we vary from these conventions. In such a case, a nearby REM statement or the program's introductory article will clearly note the special instructions.

Be sure to read each program's explanatory article — it may contain special, important information about the program. Also, use **SWAT** on your program, and get the free reprint if you don't have **SWAT**.

---

## 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.)

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 below, and mail it along with \$19.95 to **SoftSide**, 6 South Street, Milford, New Hampshire 03055.

## The Best of SoftSide

☐ **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®	<input type="text"/>	<input type="text"/>
Atari®	<input type="text"/>	<input type="text"/>
TRS-80® Mod I	<input type="text"/>	<input type="text"/>
TRS-80® Mod III	<input type="text"/>	<input type="text"/>
<b>TOTAL COPIES</b>	@ \$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

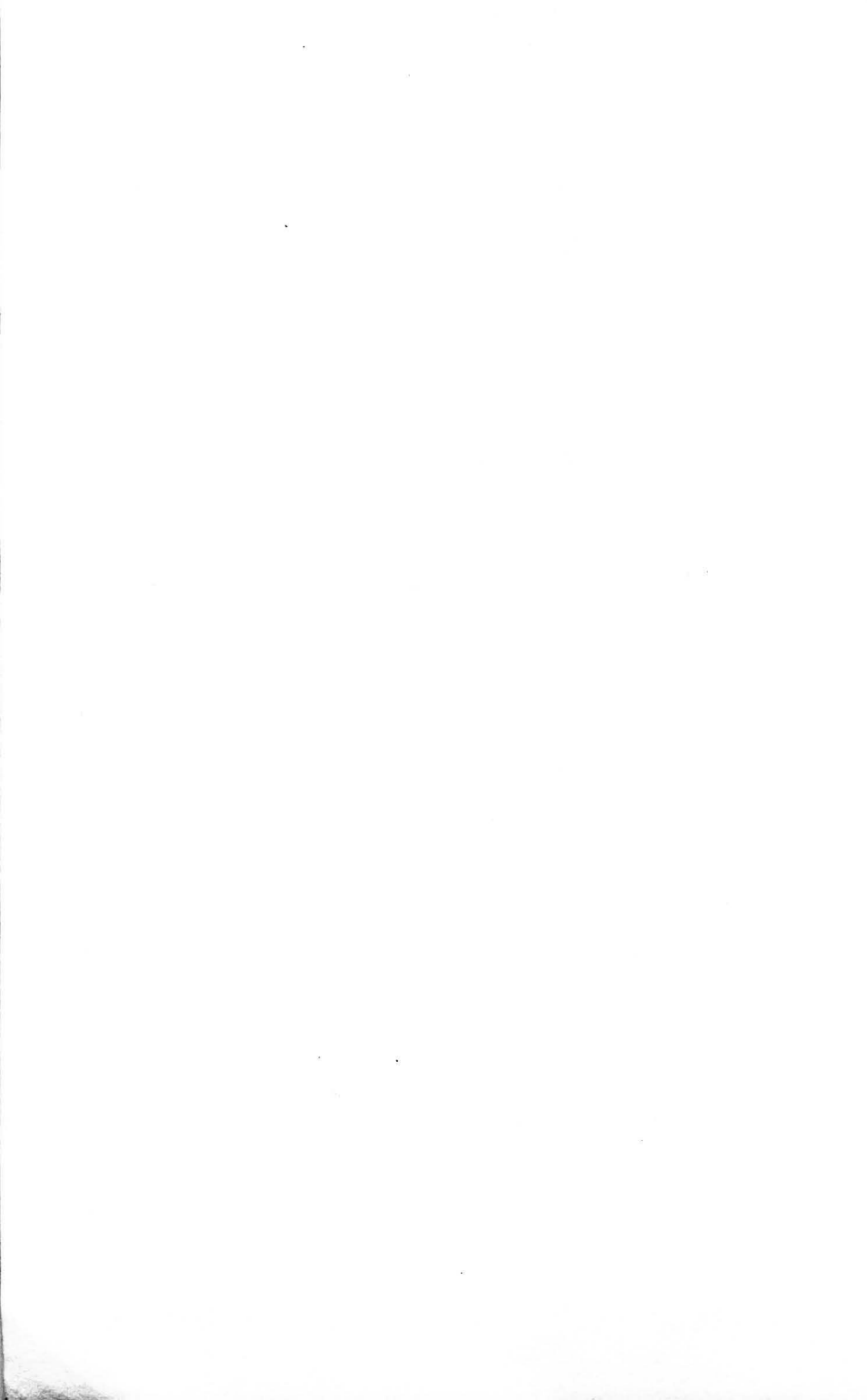
(Foreign order please include \$5 postage PAY IN USA FUNDS ONLY.)



VISA/MasterCard# and Interbank#  Exp. Date

Name of Cardholder

Signature





# SoftSide<sup>TM</sup> Selections

Here's **SoftSide Selections**, the handy, pull-out booklet with program listings for your Atari 400/800/1200 computer. This issue, **SoftSide Selections** for the Atari features:

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

- **Atari Blaster** — Alone with your computer in a space station, you must defend the Atari against nasty aliens who want to steal it...

- **Atari DV Bonus Program: Five Card Stud Poker** simulates a group of card sharks who are out to win your money. Each computerized player has his own "personality" and strategy. Can you outbluff this gang?

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

---

SOFTSIDE PUBLICATIONS, INC.

Milford, New Hampshire