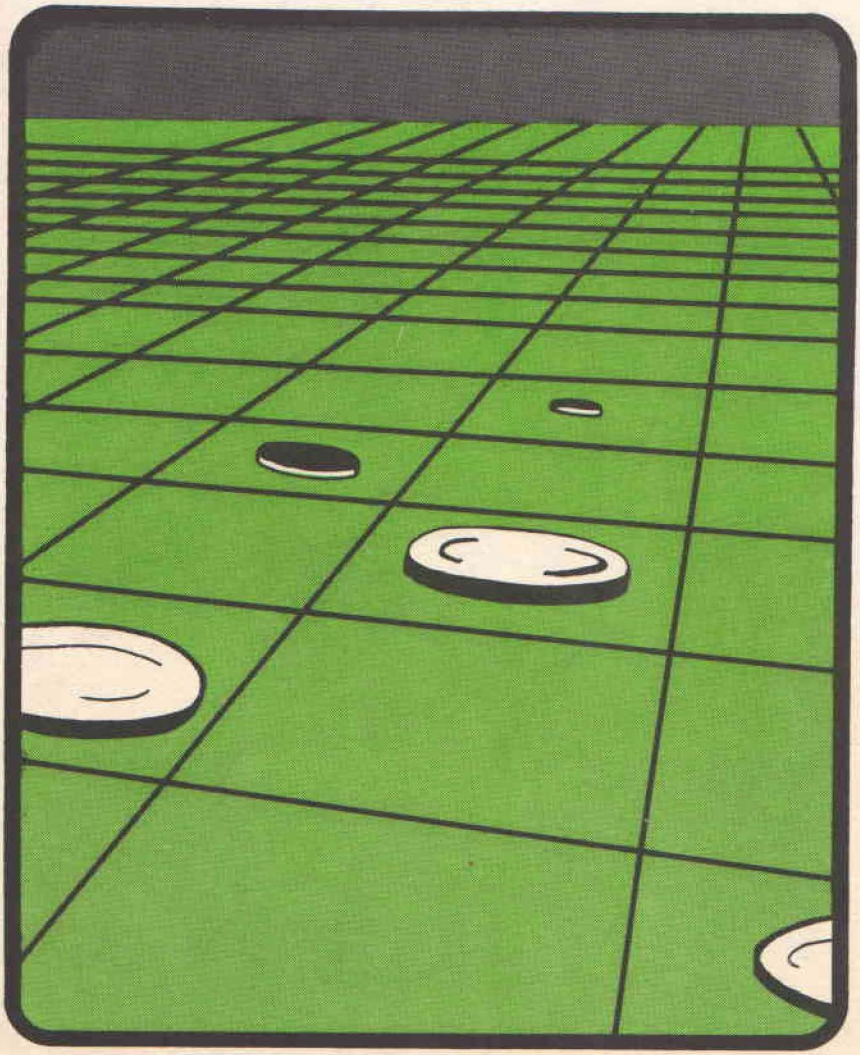
$2.95

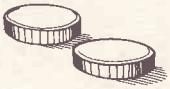
FLIP-IT II

Printed Game Software for Atari®, TRS-80®, Apple®, IBM® PC.



A product of **SoftSide** Publications, Inc.
6 South Street, Milford, NH 03055.

Index


Flip-It II Instructions 1

IBM® PC Version 4

Apple® Version 8

TRS-80® Version 16

Atari® Version 21

General Information 28

FLIP-IT II

Apple Integer BASIC original by Michael Prescott

Flip-It is a computerized version of Reversi. IBM® PC, TRS-80® and Atari® versions by Alan J. Zett. Applesoft® version by Kerry Shetline.

Flip-It II is a computerized board game in which you and the computer match wits to outflank and capture one another's pieces on an eight-by-eight board. The computer is a formidable opponent, and you won't find it a trivial matter to win.

The game begins with a square arrangement of four chips in the center of the board, two of them yours, and two belonging to the computer. You first choose a color and determine who will play first. You may also set up your own board (great for practicing strategies).

The game's object is to place one of your chips in an unoccupied square so that it outflanks one or more of the computer's chips, i.e. surround the computer chips with one of your existing chips and the new one you're playing, in a straight line. When you accomplish this, all the computer's outflanked pieces become yours. This can happen in more than one direction. In any given turn, you might capture pieces horizontally, vertically, and diagonally, resulting in a substantial shift in the relative number of chips in one turn. The outcome of the game is rarely certain until the last few moves.

The method of entering your move varies. The Apple original uses paddle zero to position a cursor on the board, and button zero to register your move. The Atari version uses the joystick to move the cursor horizontally, vertically, or diagonally from its current location, and again, pressing the button enters the move. In the TRS-80 and IBM-PC versions, all the squares are numbered, and you type the number of the square you want. The computer always checks to see that your move is legal and doesn't allow cheating.

During your turn, you may ask the computer to recommend your best move by pressing "B". If you see no possible move, press "N"; the computer then checks to see if you really have no possible move. If so, it continues with its play. If the computer has no move, it passes, after searching all the open squares. (The Atari version indicates "I pass" with a "sighing" sound.) If neither of you has a possible move (which occasionally happens even before the board is filled), the game is over and the player with the greater number of chips is the winner. You may also press "Q" to quit at any time.

With the IBM-PC and TRS-80 versions, you may type "R" to redraw the game board. This is useful if you accidentally halt execution of the game. Simply type the CONT command, then press "R". These two versions also allow the "C" option to change the game board during play. The "C" option works just like the set-up-game routine.

Winning a game involves more than capturing as many pieces as you can on any given turn. Much more important to the eventual result is the position of your chips on the board. Capturing edge and (especially) corner squares, and preventing the computer from doing the same, pays off in the long run, even if it means outflanking only one square when you could get more elsewhere on the board.

Variables

Apple version

A: Paddle position.
A(*): Decision-making array for computer's moves.
A$: Data manipulator.
B: Decision pointer.
C: Capture pointer.
CA: Capture loop variable.
CHIP: On-the-fly token.
CM: Key input variable.
COLR: Sampler variable.
D: Loop variable.
IP: "I pass" flag.
OC: Token for computer's chip.
OP: Color of computer's chips.
P1: Number of chips captured.
P2: Number of chips on board.
P4: Sum of chips captured.
Q: Substitution variable.
SY: Number of your chips on board.
X,Y,Z: Loop variables.
X1,Y1: Point under consideration.
X2,X3: Check loop variables.
XX,YY: Plot variables.
YC: Token for your chip.
YOUR: Color of your chips.
YP: "You pass" flag.

Atari version

A: Set equal to PDL.
A(*): Decision-making array for computer's moves.
A$: Data manipulator.
B: Decision pointer.
C: Capture pointer.
CA: Capture loop variable.
CHIP: On-the-fly token.
CO: Key input variable.
COLR: Sampler variable.
CLS: Clear-screen code.
CRS: Location of the cursor register.
D: Loop variable.
FL: Flip-piece flag.
IP: "I pass" flag.
KEY: Location of the keyboard register.
OC: Token for computer's chip.
OP: Color of computer's chips.
P1: Number of chips captured.
P2: Number of chips on board.
P4: Sum of chips captured.
PDL: Coordinates of chip-select cursor.
Q: Substitution variable.
RESET: Constant equal to 255.
SPK: Location of the speaker register.
ST: Stick value.
SY: Number of your chips on board.
X,Y,Z: Loop variables.
X1,Y1: Point under consideration.
X2,X3: Check loop variables.
XX,YY: Plot variables.
YC: Token for your chip.
YOUR: Color of your chips.
YP: "You pass" flag.

TRS-80 and IBM-PC Versions

A(*): Decision-making array for computer's moves.
A$: General purpose (mainly INPUTs).
B(*): Array representing board layout. (Contents: 9 represents off-board, 0 is unoccupied square, 1 is occupied by white, -1 is occupied by black.)
B$: Graphics string for a black piece.

BL: Black piece value in board array (=-1).
BP: Number of black pieces on board.
C: Value of the color (black=-1, white=1) calling the "legal check" routine.
F: At start of game, F=1 if you choose to go first. During game, used as a flag to flip pieces.
FL: If FL=0, then legal check will not flip pieces even if a legal square was found.
IP: "I pass" flag for computer.
LG: Indicates legal move if =1.
OP: Holds the value of the computer's color (-1 or 1).
OP$: Graphics string for the computer's pieces.
P: Temporary holding variables for PDL.
P2: Total number of pieces on the board.
PDL: Position of a square under consideration (1st digit is row, 2nd is column).
RF: Return flag; if =1 then RETURN from a normal (non-GOSUB) routine.
SC: Value of the color being searched for in the "legal check" routine.
W$: Graphics string for a white piece.
WH: White piece value for use in the board array (=1).
WP: Number of white pieces on the board.
X, Y, Z: Miscellaneous.
YO: Holds the value of a player's color (-1 or 1).
YO$: Graphics string for your pieces.
YP: "You pass" flag.

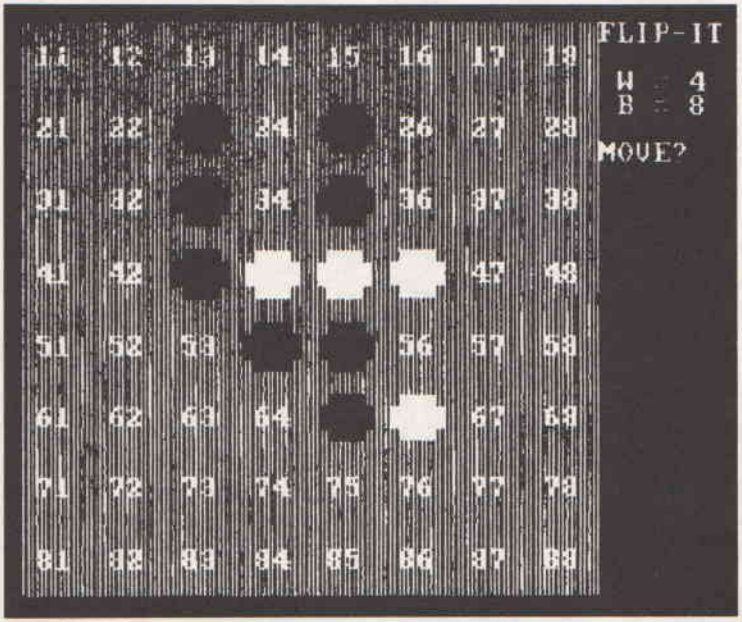


Figure 1. IBM® PC Flip-It screen.

```
SS SS SS SS SS SS SS SS SS SS SS
SS                              SS
SS          IBM PC BASIC        SS
SS         'FLIP-IT II'         SS
SS  AUTHOR: MICHAEL PRESCOTT    SS
SS     TRANSL: 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 available on #39 SoftSide DV.

Null out the function keys.

```
10 KEY OFF:FOR X=1 TO 10:KEY X,"":NEXT
```

Initialization.

```
20 SCREEN 0,1:COLOR 7,0,0:CLS:CLEAR:DEFI
NT A-Z:DIM A(64),B(99):A$="11B1188831611
383168638683363366641511484158548584353 3
4643565465642522474257547573262237326763
7672171128217872878227227774455 5445"
30 DEF SEG=0:Z=1:FOR X=1 TO 128 STEP 2:A
(Z)=ASC(MID$(A$,X,1))-48+(ASC(MID$(A$,X+
1,1))-48)*10:Z=Z+1:NEXT:FOR X!=100 TO 90
0:SOUND X!,.01:NEXT:SOUND 55,0:GOSUB 760
:CLS
```

Initialize game board and graphic strings.

```
40 FOR X=0 TO 99:B(X)=0:NEXT:B(45)=-1:B(
54)=-1:B(44)=1:B(55)=1:BP=2:WP=2:P2=4
50 PDL=0:B=0:FL=0:P2=0:RF=0:BL=-1:WH=1:O
P=0:YO=0:FOR X=0 TO 9:B(X)=9:B(X+90)=9:N
EXT:FOR X=9 TO 89 STEP 10:B(X)=9:B(X+1)=
9:NEXT
60 P1$=" "+STRING$(2,220)+" ":P2$=CHR$(2
22)+STRING$(2,219)+CHR$(221):P3$=" "+STR
ING$(2,223)+" ":B$="    "
```

Get choice of color, who goes first, and whether or not a normal board is to be used. Also set up players' pieces.

```
70 CLS:F=0:LOCATE 2,13:FOR X=1 TO 2:FOR
Y=600 TO 700 STEP 10:SOUND Y,.5:NEXT:NEX
T:POKE 1047,96:PRINT"F L I P - I T":LOCA
TE 7,1:PRINT"WHICH COLOR DO YOU WANT? (W
/B)":IF INPUT$(1)="B" THEN 90
80 YO$=W$:OP$=B$:YO=WH:OP=BL:GOTO 100
90 YO$=B$:OP$=W$:YO=BL:OP=WH
100 PRINT:PRINT"DO YOU WANT TO GO FIRST?
 (Y/N)":IF INPUT$(1)<>"N" THEN F=1
110 PRINT:PRINT"WANT TO SET UP YOUR OWN
GAME? (Y/N)":A$=INPUT$(1):CLS:GOSUB 120:
IF A$="Y" THEN B(44)=0:B(45)=0:B(54)=0:B
(55)=0:WP=0:BP=0:P2=0:GOTO 220 ELSE IF F
=1 THEN 300 ELSE GOSUB 130:GOTO 370
```

Draw game board.

```
120 COLOR 7,2:FOR X=1 TO 8:FOR Y=1 TO 8:
AZ=Y*4-3:LOCATE X*3-2,AZ:PRINT B$;:LOCAT
E X*3-1,AZ:PRINT " ";RIGHT$(STR$(X),1);M
ID$(STR$(Y),2);" ";:LOCATE X*3,AZ:PRINT
B$;:SOUND X*100+Y*50,.5:NEXT Y:NEXT X:CO
LOR 7,0:RETURN
```

Plot pieces and update game display data.

```
130 P2=0:WP=0:BP=0:FOR X=0 TO 99:IF B(X)
<>0 AND B(X)<>9 THEN Y=INT(X/10)*3:AZ=(X
 MOD 10)*4-3 ELSE 160
140 IF B(X)=1 THEN COLOR 7,2:LOCATE Y-2,
AZ:PRINT P1$;:LOCATE Y-1,AZ:PRINT P2$;:L
OCATE Y,AZ:PRINT P3$;:WP=WP+1:P2=P2+1
150 IF B(X)=-1 THEN COLOR 0,2:LOCATE Y-2
,AZ:PRINT P1$;:LOCATE Y-1,AZ:PRINT P2$;:
LOCATE Y,AZ:PRINT P3$;:BP=BP+1:P2=P2+1
160 NEXT:COLOR 7,0:LOCATE 1,33:PRINT"FLI
P-IT";:LOCATE 3,34:PRINT"W =";WP;:LOCATE
 4,34:PRINT"B =";BP;:RETURN
```

No-possible-move command.

```
170 LOCATE 6,33:PRINT"CHECKING";:FOR Z=1
 TO 60:PDL=A(Z):P=PDL:C=YO:SC=OP:FL=0:GO
SUB 480:IF LG=1 THEN 180 ELSE NEXT:IF IP
=1 THEN 410 ELSE YP=1:GOTO 370
180 LOCATE 6,33:PRINT"?WHAT   ";:LOCATE
7,33:PRINT"ABOUT  ";:A$=LEFT$(STR$(PDL),
2)+" "+RIGHT$(STR$(PDL),1):FOR X=1 TO 20
:LOCATE 8,33:PRINT A$;:FOR Y=0 TO 99:NEX
T:LOCATE 8,34:PRINT"        ";:FOR Y=0 TO
 99:NEXT:NEXT:GOTO 310
```

Find player's best move.

```
190 LOCATE 6,33:PRINT"LOOKING ";:FOR Z=1
 TO 60:PDL=A(Z):P=PDL:C=YO:SC=OP:FL=0:GO
SUB 480:IF LG=1 THEN 200 ELSE NEXT:IF IP
=1 THEN 410 ELSE YP=1:GOTO 370
200 LOCATE 7,33:PRINT"LOOKS   ";:LOCATE
8,33:PRINT"GOOD!";:A$=LEFT$(STR$(PDL),2)
+" "+RIGHT$(STR$(PDL),1):FOR X=1 TO 20:L
OCATE 6,33:PRINT A$;:FOR Y=0 TO 99:NEXT:
LOCATE 6,33:PRINT"        ";:FOR Y=0 TO 9
9:NEXT:NEXT:LOCATE 8,33:PRINT"      ";:GO
TO 310
```

Quit-game routine.

```
210 CLS:LOCATE 2,13:PRINT"F L I P - I T"
:PRINT:PRINT:PRINT"DO YOU REALLY WANT TO
 QUIT? (Y/N)":IF INPUT$(1)<>"Y" THEN CLS
:GOSUB 120:GOTO 300 ELSE 420
```

Set up your own board.

```
220 RF=1:LOCATE 8,33:PRINT"(D=DONE)";:GO
SUB 310:IF A$="D" THEN 240 ELSE LOCATE 6
,33:PRINT"COLOR?  ";
230 A$=INPUT$(1):SOUND 999,.1
240 IF A$="D" THEN FOR X=6 TO 8:LOCATE X
,33:PRINT"        ";:NEXT:GOSUB 850:P2=W
H:WP=WH:BP=WH:IF F=1 THEN 300 ELSE GOSUB
 130:GOTO 370
250 Y=INT(PDL/10)*3:AZ=(PDL MOD 10)*4-3:
LOCATE Y-2,AZ
260 IF A$="W" THEN COLOR 7,2:PRINT P1$;:
LOCATE Y-1,AZ:PRINT P2$;:LOCATE Y,AZ:PRI
NT P3$;:B(PDL)=WH:COLOR 7,0
270 IF A$="B" THEN COLOR 0,2:PRINT P1$;:
LOCATE Y-1,AZ:PRINT P2$;:LOCATE Y,AZ:PRI
NT P3$;:B(PDL)=BL:COLOR 7,0
280 IF A$=CHR$(27) THEN COLOR 7,2:PRINT
B$;:LOCATE Y-1,AZ:PRINT PDL;:LOCATE Y,AZ
:PRINT B$;:B(PDL)=0:COLOR 7,0
290 GOTO 220
```

Get players 's move.

```
300 IF P2=64 OR WP=0 OR BP=0 THEN 410 EL
SE GOSUB 130:IF P2=64 OR WP=0 OR BP=0 TH
EN 410 ELSE SOUND 666,1:SOUND 32767,1:SO
UND 666,1:POKE 1047,96
310 PDL=0:YP=0:LOCATE 7,33:PRINT"
";:LOCATE 6,33:IF RF=1 THEN PRINT"SQUARE
? "; ELSE PRINT"MOVE?   ";
320 FOR X=0 TO 1
330 A$=INPUT$(1):SOUND 999,.1:IF A$="C"
AND RF=0 THEN F=1:GOTO 220 ELSE IF A$="R
" AND RF=0 THEN CLS:GOSUB 120:GOSUB 130:
GOTO 310 ELSE IF A$="Q" AND RF=0 THEN 21
0 ELSE IF A$="N" AND RF=0 THEN 170 ELSE
IF A$="B" AND RF=0 THEN 190
340 IF A$="D" AND RF=1 THEN RF=0:RETURN
ELSE IF A$<"1" OR A$>"8" THEN LOCATE 7,3
3:PRINT"INVALID";:FOR X=222 TO 999 STEP
11:SOUND X,.01:SOUND 1111-X,.01:NEXT:SOU
ND 55,0:GOTO 310
350 LOCATE 7,34+X*2:PRINT A$;:PDL=PDL+VA
L(A$)*10^(1-X):NEXT:IF RF=1 THEN RF=0:RE
TURN
360 C=YO:SC=OP:FL=1:GOSUB 480:IF LG=0 TH
EN A$="O":GOTO 340 ELSE GOSUB 130
```

Choose computer's move.

```
370 IF P2=64 OR WP=0 OR BP=0 THEN 410 EL
SE SOUND 666,1:SOUND 32767,1:SOUND 666,1
:POKE 1047,96
380 IP=0:LOCATE 6,33:PRINT"MAYBE   ";:FO
R Z=1 TO 64:SOUND 37,.1:PDL=A(Z):LOCATE
7,33:PRINT LEFT$(STR$(PDL),2)" "RIGHT$(S
TR$(PDL),1);:P=PDL:IF B(P)<>0 THEN 390 E
LSE C=OP:SC=YO:FL=1:GOSUB 480:IF LG<>0 T
HEN 880
390 NEXT Z
```

PC

PC

Computer must pass.

```
400 IF P2<>64 AND YP=0 THEN LOCATE 9,33:
PRINT"PASS!";:IP=1:FOR X=222 TO 999 STEP
 11:SOUND X,.01:SOUND 1111-X,.01:NEXT:SO
UND 55,0:LOCATE 9,33:PRINT"      ";:GOTO
300
```

End-game routine.

```
410 LOCATE 9,33:PRINT"NO MOVES";:LOCATE
10,35:PRINT"LEFT!";:LOCATE 12,33:PRINT"$
 GAME $";:LOCATE 13,33:PRINT"$ OVER $";:
FOR X=0 TO 900 STEP 3:SOUND 1000-X,.7:SO
UND 1500-X,.7:IF INKEY$="" THEN NEXT
420 CLS:LOCATE 2,13:PRINT"F L I P - I T"
:PRINT:PRINT:PRINT"FINAL SCORE:":PRINT:P
RINT"    WHITE ="WP:PRINT"    BLACK ="BP
:PRINT
430 IF WP=BP THEN PRINT"CONGRATULATIONS.
 IT'S A DRAW.":PRINT:PRINT"WE APPEAR TO
BE EVENLY MATCHED.":PRINT:GOTO 460 ELSE
IF (WP>BP AND OP=WH) OR (BP>WP AND OP=BL
) THEN PRINT"THANK YOU FOR A STIMULATING
 GAME.":PRINT:PRINT"I HAVE WON THIS GAME
 BY";:GOTO 450
440 PRINT"CONGRATULATIONS!":PRINT:PRINT"
YOU HAVE WON THIS GAME BY";
450 PRINT ABS(WP-BP)"CHIPS.":PRINT
```

Check for new game.

```
460 PRINT"WOULD YOU LIKE TO PLAY AGAIN?
(Y/N)":IF INPUT$(1)<>"N" THEN CLS:GOTO 4
0
470 CLEAR:PRINT:PRINT"GOOD BYE...":FOR X
=0 TO 4:SOUND 555,.7:SOUND 32767,.3:NEXT
:KEY ON:POKE 1047,64:DEF SEG:GOTO 870
```

Legal check routine. Combines a search for legal moves with a routine to flip pieces.

```
480 LG=0:IF B(PDL)<>0 THEN 750
490 FOR Y=1 TO 8:F=0:ON Y GOTO 500,530,5
60,590,620,650,680,710
500 P=PDL:FOR X=1 TO 8:P=P+10:IF P>99 TH
EN 740 ELSE IF B(P)=0 OR (B(P)=C AND X=1
) OR B(P)=9 THEN 740 ELSE IF B(P)=SC THE
N IF F=1 THEN B(P)=C:GOTO 520 ELSE 520 E
LSE F=1:LG=1:P=PDL:IF FL=0 THEN 750 ELSE
 B(P)=C:GOTO 500
510 GOTO 530
520 NEXT X:GOTO 740
530 P=PDL:FOR X=1 TO 8:P=P-10:IF P<0 THE
N 740 ELSE IF B(P)=0 OR (B(P)=C AND X=1)
 OR B(P)=9 THEN 740 ELSE IF B(P)=SC THEN
 IF F=1 THEN B(P)=C:GOTO 550 ELSE 550 EL
SE F=1:LG=1:P=PDL:IF FL=0 THEN 750 ELSE
B(P)=C:GOTO 530
540 GOTO 560
550 NEXT X:GOTO 740
560 P=PDL:FOR X=1 TO 8:P=P+1:IF P>99 THE
N 740 ELSE IF B(P)=0 OR (B(P)=C AND X=1)
 OR B(P)=9 THEN 740 ELSE IF B(P)=SC THEN
 IF F=1 THEN B(P)=C:GOTO 580 ELSE 580 EL
SE F=1:LG=1:P=PDL:IF FL=0 THEN 750 ELSE
B(P)=C:GOTO 560
570 GOTO 590
580 NEXT X:GOTO 740
590 P=PDL:FOR X=1 TO 8:P=P-1:IF P<0 THEN
 740 ELSE IF B(P)=0 OR (B(P)=C AND X=1)
OR B(P)=9 THEN 740 ELSE IF B(P)=SC THEN
IF F=1 THEN B(P)=C:GOTO 610 ELSE 610 ELS
E F=1:LG=1:P=PDL:IF FL=0 THEN 750 ELSE B
(P)=C:GOTO 590
600 GOTO 620
610 NEXT X:GOTO 740
620 P=PDL:FOR X=1 TO 8:P=P+9:IF P>99 THE
N 740 ELSE IF B(P)=0 OR (B(P)=C AND X=1)
 OR B(P)=9 THEN 740 ELSE IF B(P)=SC THEN
 IF F=1 THEN B(P)=C:GOTO 640 ELSE 640 EL
SE F=1:LG=1:P=PDL:IF FL=0 THEN 750 ELSE
B(P)=C:GOTO 620
630 GOTO 650
640 NEXT X:GOTO 740
650 P=PDL:FOR X=1 TO 8:P=P-9:IF P<0 THEN
 740 ELSE IF B(P)=0 OR (B(P)=C AND X=1)
OR B(P)=9 THEN 740 ELSE IF B(P)=SC THEN
IF F=1 THEN B(P)=C:GOTO 670 ELSE 670 ELS
E F=1:LG=1:P=PDL:IF FL=0 THEN 750 ELSE B
(P)=C:GOTO 650
660 GOTO 680
670 NEXT X:GOTO 740
680 P=PDL:FOR X=1 TO 8:P=P+11:IF P>99 TH
EN 740 ELSE IF B(P)=0 OR (B(P)=C AND X=1
) OR B(P)=9 THEN 740 ELSE IF B(P)=SC THE
N IF F=1 THEN B(P)=C:GOTO 700 ELSE 700 E
LSE F=1:LG=1:P=PDL:IF FL=0 THEN 750 ELSE
 B(P)=C:GOTO 680
690 GOTO 710
```

```
700 NEXT X:GOTO 740
710 P=PDL:FOR X=1 TO 8:P=P-11:IF P<0 THE
N 740 ELSE IF B(P)=0 OR (B(P)=C AND X=1)
 OR B(P)=9 THEN 740 ELSE IF B(P)=SC THEN
 IF F=1 THEN B(P)=C:GOTO 730 ELSE 730 EL
SE F=1:LG=1:P=PDL:IF FL=0 THEN 750 ELSE
B(P)=C:GOTO 710
720 GOTO 740
730 NEXT X
740 FOR X=0 TO 1:NEXT X:NEXT Y
750 FOR X=0 TO 1:NEXT X:FOR Y=0 TO 1:NEX
T Y:RETURN
```

Game Instructions.

```
760 LOCATE 12,1,0:PRINT"NEED INSTRUCTION
S? (Y/N)":IF INPUT$(1)<>"Y" THEN RETURN
770 CLS:LOCATE 2,13:PRINT"F L I P - I T"
:PRINT:PRINT"   THE OBJECT OF THIS GAME
IS TO FILL   THE BOARD WITH AS MANY PIEC
ES OF YOUR   COLOR AS YOU CAN. TO DO THI
S YOU MUST   OUTFLANK YOUR OPPONENT'S PI
ECES AND     FLIP THEM TO YOUR COLOR."
780 PRINT:PRINT"   OUTFLANKING CAN OCCUR
 HORIZONTALLY,  VERTICALLY, OR DIAGONALL
Y.  THE GAME    ENDS WHEN THE BOARD IS F
ULL, OR WHEN    BOTH PLAYERS CAN'T MOVE.
 AT THAT TIME,  WHOEVER HAS THE MOST PIE
CES WINS."
790 LOCATE 24,5:PRINT"(PRESS <RETURN> TO
 CONTINUE)";
800 IF INKEY$<>CHR$(13) THEN 800
810 CLS:LOCATE 2,13:PRINT"F L I P - I T"
:PRINT:PRINT"   YOU CHOOSE YOUR MOVE BY
TYPING THE   ROW AND COLUMN NUMBERS COMB
INED. FOR    EXAMPLE: THE UPPER RIGHT-HA
ND CORNER    IS THE  POSITION 18 - THAT
IS; 1 FOR    THE FIRST ROW AND 8 FOR THE
 EIGHTH"
820 PRINT"COLUMN. IF YOU MAKE A MISTAKE
TYPING A  NUMBER, TYPE A NINE FOR EITHER
 OF THE   NUMBERS AND IT WILL MARK THE E
NTRY AS   INVALID. DURING THE GAME YOU A
LSO HAVE  THE FOLLOWING OPTIONS:"
830 PRINT:PRINT:PRINT"Q - TO QUIT THE GA
ME":PRINT"R - REDRAW THE GAME BOARD":PRI
NT"N - NO AVAILABLE MOVE":PRINT"C - CHAN
GE GAME BOARD SETUP":PRINT"B - ASK FOR T
HE BEST MOVE":LOCATE 24,5:PRINT"(PRESS <
RETURN> TO START GAME)";
840 IF INKEY$<>CHR$(13) THEN 840 ELSE RE
TURN
```

Print game board over players' pieces.

```
850 FOR X=1 TO 8:FOR Y=1 TO 8:COLOR 7,2:
AZ=B(X*10+Y):IF AZ=1 THEN COLOR 0,7 ELSE
 IF AZ=-1 THEN COLOR 7,0
860 LOCATE X*3-1,Y*4-2:PRINT RIGHT$(STR$
(X),1);MID$(STR$(Y),2);:NEXT Y:NEXT X:CO
LOR 7,0:RETURN
```

Restore default function keys.

```
870 KEY 1,"LIST ":KEY 2,"RUN"+CHR$(13):K
EY 3,"LOAD"+CHR$(34):KEY 4,"SAVE"+CHR$(3
4):KEY 5,"CONT"+CHR$(13):KEY 6,","+CHR$(
34)+"LPT1:"+CHR$(13):KEY 7,"TRON"+CHR$(1
3):KEY 8,"TROFF"+CHR$(13):KEY 9,"KEY ":K
EY 10,"SCREEN 0,0,0"+CHR$(13):POKE 1047,
0:END
```

Flash the number of the computer's choice.

```
880 COLOR 23,2:LOCATE INT(PDL/10)*3-1,(P
DL MOD 10)*4-3:PRINT PDL;:FOR X=0 TO 400
0:NEXT:COLOR 7,0:GOTO 300
```

For IBM® PC FLIP-IT II

LINES	SWAT CODE	LENGTH
10 - 50	DM	519
60 - 110	PC	505
120 - 170	OP	592
180 - 210	JB	549
220 - 300	BS	579
310 - 360	RE	510
370 - 420	KP	582
430 - 490	UB	521
500 - 560	RW	568
570 - 650	MC	589
660 - 760	ZG	527
770 - 810	VR	719
820 - 860	MA	543
870 - 880	MB	278

PC

```
SS SS SS SS SS SS SS SS SS SS SS
SS                              SS
SS      APPLESOFT  BASIC        SS
SS         'FLIP-IT II'         SS
SS  AUTHOR: MICHAEL PRESCOTT    SS
SS   TRANSL: KERRY SHETLINE     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 #39 SoftSide CV and DV.

Initialization.

```
10  DIM A(60):Z = 1: GOSUB 2150: FOR
    X = 1 TO  LEN (A$) STEP 2:A(
    Z) =  ASC ( MID$ (A$,X,1)) -
    48 + ( ASC ( MID$ (A$,X + 1,
    1)) - 48) * 10:Z = Z + 1: NEXT
    X: GOSUB 1960
20 G$ =  CHR$ (7): DEF  FN M2(X) =
    X -  INT (X / 2) * 2: DEF  FN
    M8(X) = X -  INT (X / 8) * 8
    : DEF  FN MT(X) = X -  INT (
    X / 10) * 10
30  GOSUB 2160: FOR X = 1 TO  LEN
    (A$) STEP 2:A(Z) =  ASC ( MID$
    (A$,X,1)) - 48 + ( ASC ( MID$
    (A$,X + 1,1)) - 48) * 10:Z =
    Z + 1: NEXT X
40  POKE  - 16368,0:B = C = D:YOU
    R = 0:OP = 15:P2 = 4:SY = 2
```

Graphic introduction.

```
50  TEXT : GR : HOME : HTAB 15: PRINT
    "FLIP-IT ]" CHR$ (91): POKE
    34,21
60  POKE 49168,0: FOR Y = 1 TO 22
    : COLOR= Y: IF  FN M2(Y) = 0
     THEN 80
70  FOR X = 0 TO 19: VLIN 0,39 AT
    X: VLIN 0,39 AT 39 - X: NEXT
    X: GOTO 100
80  FOR X = 0 TO 19: HLIN 0,39 AT
    X: HLIN 0,39 AT 39 - X: IF  PEEK
    (49152) > 127 THEN 110
90  NEXT X
100  NEXT Y
110  POKE 49168,0: COLOR= 6: FOR
    X = 0 TO 19: VLIN 0,39 AT X:
     VLIN 0,39 AT 39 - X: NEXT X
    : COLOR= 12: FOR X = 1 TO 7:
     VLIN 0,39 AT 5 * X - 1: HLIN
    0,39 AT 5 * X - 1: NEXT X
```

Get choice of color.

```
120  HOME : PRINT "WHICH COLOR WO
    ULD YOU LIKE (W,B)? ";
130  GET A$: IF A$ = "B" THEN 160

140  IF A$ <  > "W" THEN 130
150 YOUR = 15:OP = 0:YC = 32:OC =
    48: GOTO 170
160 YOUR = 0:OP = 15:YC = 48:OC =
    32
170 XX = 2:YY = 2: COLOR= 9: PLOT
    XX,YY:COLR = 6
```

Prompt for type of game set-up.

```
180  HOME : PRINT "DO YOU WANT TO
     SET UP YOUR OWN GAME? ";
190  GET A$: IF A$ = "Y" THEN 166
    0
200  IF A$ <  > "N" THEN 190
```

Set up new game.

```
210  COLOR= 0:X = 45:CHIP = 48: GOSUB
    1040:X = 54: GOSUB 1040: COLOR=
    15:X = 44:CHIP = 32: GOSUB 1
    040:X = 55: GOSUB 1040
```

Print options.

```
220  POKE 34,21: HOME : PRINT "Q-
     QUIT  N-NO POSSIBLE MOVE  B-
     BEST MOVE": POKE 34,22
```

Decide who goes first.

```
230  HOME : PRINT "DO YOU WANT TO
      GO FIRST? ";
240  GET A$: IF A$ = "N" THEN 530

250  IF A$ < > "Y" THEN 240
```

If board is filled, then end; otherwise update the chip-selector cursor.

```
260  IF P2 = 64 THEN 1160
```

Get command, or update chip-selector cursor.

```
270  GOSUB 2170: GOTO 470
280  IF  PEEK ( - 16384) < 128 THEN
     460
290 CM =  PEEK ( - 16384): POKE  -
     16368,0: IF CM < > 209 AND
     CM < > 206 AND CM < > 194 THEN
     460
300  IF CM = 209 THEN 340
310  IF CM = 206 THEN 380
320  IF CM = 194 THEN  GOSUB 400
330  GOTO 460
```

End-game command.

```
340  POKE 34,22: HOME : PRINT "DO
      YOU WANT TO END THE GAME? "
     ;
350  GET A$: IF A$ = "N" THEN  HOME
     : GOTO 460
360  IF A$ < > "Y" THEN 350
370  TEXT : HOME : END
```

No-possible-move command.

```
380  PRINT : PRINT "LET ME CHECK.
     ";: GOSUB 400: IF YP = 1 THEN
     530
390  PRINT : PRINT "I FOUND ONE!"
     ;: FOR X = 1 TO 600: NEXT X:
      HOME : GOTO 460
```

Find best move.

```
400 FL = 1:Q = YOUR:YOUR = OP:OP =
     Q: GOSUB 1060: IF YP = 1 THEN
     450
410  FOR CA = 1 TO 30: IF  FN M2(
     CA) = 0 THEN  COLOR= 9
420  IF  FN M2(CA) THEN  COLOR= 6

430  IF CA -  INT (CA / 5) * 5 THEN
      GOSUB 1040
440  NEXT CA
450 Q = YOUR:YOUR = OP:OP = Q:FL =
     0: RETURN
```

Position and display player's move.

```
460  IF A =  INT (63 *  PDL (0) /
     255) THEN 480
470 A =  INT (63 *  PDL (0) / 255
     ): COLOR= COLR: PLOT XX,YY: COLOR=
9:XX = 5 *  FN MB(A) + 2:YY =  INT
     (A / 8) * 5 + 2:COLR =  SCRN(
     XX,YY - 1): PLOT XX,YY
480  IF  PEEK ( - 16287) < 128 THEN
     280
490  IF COLR < > 6 THEN 280
500 P3 = SY:CHIP = YC: COLOR= YOU
     R:P4 = 0:D = 0: FOR CA = 1 TO
     8:X =  FN MB(A) + 1 + 10 * (
      INT (A / 8) + 1):P1 = 0: ON
     CA GOSUB 600,650,700,750,800
     ,860,920,980:P4 = P4 + P1: IF
     P1 THEN D = D + 1
510  NEXT CA: IF D = 0 THEN 280
520 SY = SY + P4 - D + 1:P2 = P2 +
     1
```

Do computer's move.

```
530  GOSUB 2170: GOSUB 1060: FOR
     CA = 1 TO 90: IF  FN M2(CA) =
     0 THEN  COLOR= 9
540  IF  FN M2(CA) THEN  COLOR= 6
```

```
550  IF CA / 7 =  INT (CA / 7) THEN
      GOSUB 1040
560  NEXT CA: COLOR= OP
570 CHIP = OC:P2 = P2 + 1:Q = YOU
     R:YOUR = OP:OP = Q:P4 = 0:D =
     0: FOR CA = 1 TO 8:X = A(B):
     P1 = 0: ON CA GOSUB 600,650,
     700,750,800,860,920,980: IF
     P1 THEN D = D + 1
580 P4 = P4 + P1: NEXT CA:SY = SY
      - P4 + D
590 Q = YOUR:YOUR = OP:OP = Q:COL
     R =  SCRN( XX,YY - 1): GOTO
     260
```

Capture routines.

```
600 X1 = 5 * FN MT(X) - 3:Y1 =  INT
     (X / 10) * 5 - 3: FOR X2 = 1
      TO 7:Y1 = Y1 - 5: IF Y1 < 0
      THEN  RETURN
610  IF  SCRN( X1,Y1) = YOUR AND
     X2 = 1 OR  SCRN( X1,Y1) = 6 THEN
      RETURN
620  IF  SCRN( X1,Y1) = OP THEN  NEXT
     X2
630  IF X2 = 8 THEN  RETURN
640  GOSUB 1040:P1 = X2: FOR X3 =
     1 TO X2:X = X - 10: GOSUB 10
     40: NEXT X3: RETURN
650 X1 = 5 * FN MT(X) - 3:Y1 =  INT
     (X / 10) * 5 - 3: FOR X2 = 1
      TO 7:Y1 = Y1 + 5: IF Y1 > 3
     9 THEN  RETURN
660  IF  SCRN( X1,Y1) = YOUR AND
     X2 = 1 OR  SCRN( X1,Y1) = 6 THEN
      RETURN
670  IF  SCRN( X1,Y1) = OP THEN  NEXT
     X2
680  IF X2 = 8 THEN  RETURN
690  GOSUB 1040:P1 = X2: FOR X3 =
     1 TO X2:X = X + 10: GOSUB 10
     40: NEXT X3: RETURN
700 X1 = 5 * FN MT(X) - 3:Y1 =  INT
     (X / 10) * 5 - 3: FOR X2 = 1
      TO 7:X1 = X1 - 5: IF X1 < 0
      THEN  RETURN
710  IF  SCRN( X1,Y1) = YOUR AND
     X2 = 1 OR  SCRN( X1,Y1) = 6 THEN
     RETURN
720  IF  SCRN( X1,Y1) = OP THEN  NEXT
     X2
730  IF X2 = 8 THEN  RETURN
```

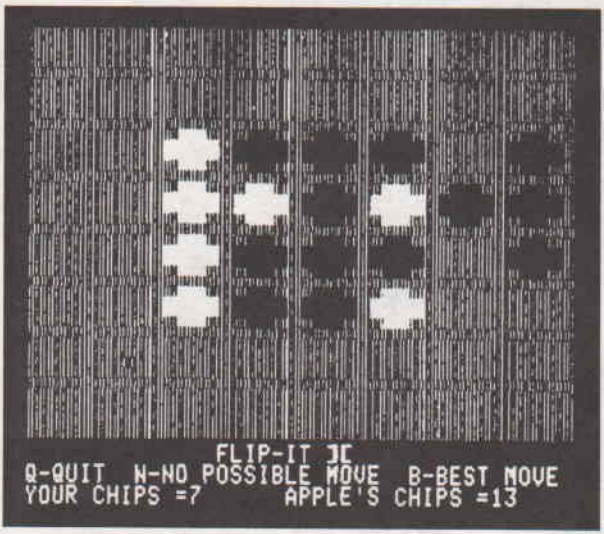


Figure 2. Apple® Flip-It screen.

```
740  GOSUB 1040:P1 = X2: FOR X3 =
     1 TO X2:X = X - 1: GOSUB 104
     0: NEXT X3: RETURN
750 X1 = 5 * FN MT(X) - 3:Y1 = INT
     (X / 10) * 5 - 3: FOR X2 = 1
      TO 7:X1 = X1 + 5: IF X1 > 3
     9 THEN RETURN
760  IF SCRN( X1,Y1) = YOUR AND
     X2 = 1 OR SCRN( X1,Y1) = 6 THEN
      RETURN
770  IF SCRN( X1,Y1) = OP THEN NEXT
     X2
780  IF X2 = 8 THEN RETURN
790  GOSUB 1040:P1 = X2: FOR X3 =
     1 TO X2:X = X + 1: GOSUB 104
     0: NEXT X3: RETURN
800 X1 = 5 * FN MT(X) - 3:Y1 = INT
     (X / 10) * 5 - 3: FOR X2 = 1
      TO 7:X1 = X1 - 5:Y1 = Y1 - 5
810  IF X1 < 0 OR Y1 < 0 THEN RETURN

820  IF SCRN( X1,Y1) = YOUR AND
     X2 = 1 OR SCRN( X1,Y1) = 6 THEN
      RETURN
830  IF SCRN( X1,Y1) = OP THEN NEXT
     X2
840  IF X2 = 8 THEN RETURN
850  GOSUB 1040:P1 = X2: FOR X3 =
     1 TO X2:X = X - 11: GOSUB 10
     40: NEXT X3: RETURN
860 X1 = 5 * FN MT(X) - 3:Y1 = INT
     (X / 10) * 5 - 3: FOR X2 = 1
      TO 7:X1 = X1 + 5:Y1 = Y1 - 5
870  IF X1 > 39 OR Y1 < 0 THEN RETURN

880  IF SCRN( X1,Y1) = YOUR AND
     X2 = 1 OR SCRN( X1,Y1) = 6 THEN
      RETURN
890  IF SCRN( X1,Y1) = OP THEN NEXT
     X2
900  IF X2 = 8 THEN RETURN
910  GOSUB 1040:P1 = X2: FOR X3 =
     1 TO X2:X = X - 9: GOSUB 104
     0: NEXT X3: RETURN
920 X1 = 5 * FN MT(X) - 3:Y1 = INT
     (X / 10) * 5 - 3: FOR X2 = 1
      TO 7:X1 = X1 + 5:Y1 = Y1 + 5
930  IF X1 > 39 OR Y1 > 39 THEN RETURN
940  IF SCRN( X1,Y1) = YOUR AND
     X2 = 1 OR SCRN( X1,Y1) = 6 THEN
      RETURN
950  IF SCRN( X1,Y1) = OP THEN NEXT
     X2
960  IF X2 = 8 THEN RETURN
970  GOSUB 1040:P1 = X2: FOR X3 =
     1 TO X2:X = X + 11: GOSUB 10
     40: NEXT X3: RETURN
980 X1 = 5 * FN MT(X) - 3:Y1 = INT
     (X / 10) * 5 - 3: FOR X2 = 1
      TO 7:X1 = X1 - 5:Y1 = Y1 + 5
990  IF X1 < 0 OR Y1 > 39 THEN RETURN

1000  IF SCRN( X1,Y1) = YOUR AND
      X2 = 1 OR SCRN( X1,Y1) = 6 THEN
       RETURN
1010  IF SCRN( X1,Y1) = OP THEN
       NEXT X2
1020  IF X2 = 8 THEN RETURN
1030  GOSUB 1040:P1 = X2: FOR X3 =
      1 TO X2:X = X + 9: GOSUB 104
      0: NEXT X3: RETURN
```

Draw chip at selected location.

```
1040 X1 = 5 * FN MT(X) - 5:Y1 =
      INT (X / 10) * 5 - 5: VLIN
      Y1 + 1,Y1 + 2 AT X1: VLIN Y1
      ,Y1 + 3 AT X1 + 1: VLIN Y1,Y
      1 + 3 AT X1 + 2: VLIN Y1 + 1
      ,Y1 + 2 AT X1 + 3
1050  FOR Z = 1 TO 3: POKE - 163
      36, PEEK ( - 16336): NEXT Z:
       RETURN
```

Step through the best moves for the computer. If none are found then pass or concede game.

```
1060  IF P2 - SY = 0 OR SY = 0 THEN
      1160
1070  IF FL = 0 THEN IP = 0
1080  IF FL = 1 THEN YP = 0
1090  FOR B = 1 TO 60:X = A(B):X1
       = 5 * FN MT(X) - 3:Y1 = INT
      (X / 10) * 5 - 3: POKE - 16
      336, PEEK ( - 16336): IF SCRN(
      X1,Y1) = 6 THEN GOSUB 1250
```

```
1100  NEXT B
1110  IF P2 - SY <  > 0 AND SY <
      > 0 AND FL = 0 AND YP = 0 THEN
      1140
1120  IF P2 - SY <  > 0 AND SY <
      > 0 AND FL = 1 AND IP = 0 THEN
      1150
1130  GOTO 1160
1140 IP = 1: VTAB 24: HTAB 15: PRINT
     G$;G$"I MUST PASS";: FOR X =
     1 TO 800: NEXT X: VTAB 24: HTAB
     1: CALL  - 958: GOTO 260
1150 YP = 1: RETURN
1160  PRINT : PRINT "NO POSSIBLE
      MOVES LEFT!";: FOR X = 1 TO
      1000: NEXT X: PRINT : IF P2 -
      SY > SY THEN 1190
1170  IF P2 - SY = SY THEN 1200
1180  PRINT "YOU WON THIS ROUND B
      Y ";2 * SY - P2;" CHIPS.";: GOTO
      1210
1190  PRINT "I WON BY "P2 - 2 * S
      Y" CHIPS.";: FOR X = 1 TO 10
      00: NEXT X: PRINT : PRINT "T
      HANK YOU FOR A STIMULATING G
      AME.";: GOTO 1210
1200  PRINT "CONGRATULATIONS. IT'
      S A DRAW.";
1210  FOR X = 1 TO 10: POKE  - 16
      336, PEEK ( - 16336): NEXT X
      : FOR X = 1 TO 1500: NEXT X:
      POKE 34,20: HOME : PRINT "D
      O YOU WANT TO PLAY AGAIN? ";

1220  GET A$: IF A$ = "Y" THEN 40

1230  IF A$ <  > "N" THEN 1220
1240  TEXT : HOME : END
```

The computer has found an empty square; now check if it is a legal move.

```
1250  FOR C = 1 TO B:X1 = 5 *  FN
      MT(X) - 4:Y1 =  INT (X / 10)
      * 5 - 3: POKE  - 16336, PEEK
      ( - 16336): ON C GOSUB 1260,
      1310,1360,1410,1460,1510,156
      0,1610: NEXT C: RETURN
```

Check legality of the computer's move. If it is legal then jump out of the loop and capture pieces.

```
1260  FOR D = 1 TO 7:Y1 = Y1 + 5:
      IF Y1 > 39 THEN  RETURN
1270  IF ( SCRN( X1,Y1) = OP AND
      D = 1) OR  SCRN( X1,Y1) = 6 THEN
      RETURN
1280  IF  SCRN( X1,Y1) = YOUR THEN
      NEXT D
1290  IF D = 8 THEN  RETURN
1300  POP : POP : GOTO 2140
1310  FOR D = 1 TO 7:Y1 = Y1 - 5:
      IF Y1 < 0 THEN  RETURN
1320  IF ( SCRN( X1,Y1) = OP AND
      D = 1) OR  SCRN( X1,Y1) = 6 THEN
      RETURN
1330  IF  SCRN( X1,Y1) = YOUR THEN
      NEXT D
1340  IF D = 8 THEN  RETURN
1350  POP : POP : GOTO 2140
1360  FOR D = 1 TO 7:X1 = X1 - 5:
      IF X1 < 0 THEN  RETURN
1370  IF ( SCRN( X1,Y1) = OP AND
      D = 1) OR  SCRN( X1,Y1) = 6 THEN
      RETURN
1380  IF  SCRN( X1,Y1) = YOUR THEN
      NEXT D
1390  IF D = 8 THEN  RETURN
1400  POP : POP : GOTO 2140
1410  FOR D = 1 TO 7:X1 = X1 + 5:
      IF X1 > 39 THEN  RETURN
1420  IF ( SCRN( X1,Y1) = OP AND
      D = 1) OR  SCRN( X1,Y1) = 6 THEN
      RETURN
1430  IF  SCRN( X1,Y1) = YOUR THEN
      NEXT D
1440  IF D = 8 THEN  RETURN
1450  POP : POP : GOTO 2140
1460  FOR D = 1 TO 7:X1 = X1 + 5:
      Y1 = Y1 + 5: IF X1 > 39 OR Y
      1 > 39 THEN  RETURN
1470  IF  SCRN( X1,Y1) = 6 OR  SCRN(
      X1,Y1) = OP AND D = 1 THEN  RETURN

1480  IF  SCRN( X1,Y1) = YOUR THEN
      NEXT D
1490  IF D = 8 THEN  RETURN
```

```
1500  POP : POP : GOTO 2140
1510  FOR D = 1 TO 7:X1 = X1 - 5:
      Y1 = Y1 + 5: IF X1 < 0 OR Y1
      > 39 THEN  RETURN
1520  IF  SCRN( X1,Y1) = 6 OR  SCRN(
      X1,Y1) = OP AND D = 1 THEN  RETURN

1530  IF  SCRN( X1,Y1) = YOUR THEN
      NEXT D
1540  IF D = 8 THEN  RETURN
1550  POP : POP : GOTO 2140
1560  FOR D = 1 TO 7:X1 = X1 - 5:
      Y1 = Y1 - 5: IF X1 < 0 OR Y1
      < 0 THEN  RETURN
1570  IF  SCRN( X1,Y1) = 6 OR  SCRN(
      X1,Y1) = OP AND D = 1 THEN  RETURN

1580  IF  SCRN( X1,Y1) = YOUR THEN
      NEXT D
1590  IF D = 8 THEN  RETURN
1600  POP : POP : GOTO 2140
1610  FOR D = 1 TO 7:X1 = X1 + 5:
      Y1 = Y1 - 5: IF X1 > 39 OR Y
      1 < 0 THEN  RETURN
1620  IF  SCRN( X1,Y1) = 6 OR  SCRN(
      X1,Y1) = OP AND D = 1 THEN  RETURN

1630  IF  SCRN( X1,Y1) = YOUR THEN
      NEXT D
1640  IF D = 8 THEN  RETURN
1650  POP : POP : GOTO 2140
```

Set up a game.

```
1660  HOME : PRINT "HOW DO YOU WA
      NT TO SET IT UP?": PRINT "1-
      REGULAR 2-MANUAL: ";:P2 = 0:
      S; = 0
1670  GET R$: IF R$ = "1" THEN 19
      50
1680  IF R$ <  > "2" THEN 1670
1690  HOME : PRINT "PRESS (ESC) W
      HEN FINISHED": PRINT "USE BU
      TTON TO SELECT SQUARE"
1700 A =  INT (63 *  PDL (0) / 25
      5): COLOR= COLR: PLOT XX,YY:
       COLOR= 9:XX = 5 *  FN MB(A)
       + 2:YY =  INT (A / 8) * 5 +
      2:COLR =  SCRN( XX,YY - 1): PLOT
      XX,YY
1710  IF  PEEK ( - 16384) < 128 THEN
      1730
1720 CM =  PEEK ( - 16384): POKE
      - 16368,0: IF CM = 155 THEN
      1800
1730  IF  PEEK ( - 16287) < 128 THEN
      1700
1740  HOME : PRINT "WHITE, BLACK,
       OR NO PIECE (W/B/N)? ";
1750  GET A$: IF A$ <  > "W" AND
      A$ <  > "B" AND A$ <  > "N" THEN
      1750
1760 X =  FN MB(A) + 1 + 10 * ( INT
      (A / 8) + 1)
1770  COLOR= 0: IF A$ = "W" THEN
       COLOR= 15
1780  IF A$ = "N" THEN  COLOR= 6
1790  GOSUB 1040: COLOR=  SCRN( X
      X,YY - 1): GOTO 1690
1800 X = 44: GOSUB 1940:Y1 = Y1 +
      1: IF  SCRN( X1,Y1) = 6 THEN
      1930
1810 X = 45: GOSUB 1940:Y1 = Y1 +
      1: IF  SCRN( X1,Y1) = 6 THEN
      1930
```

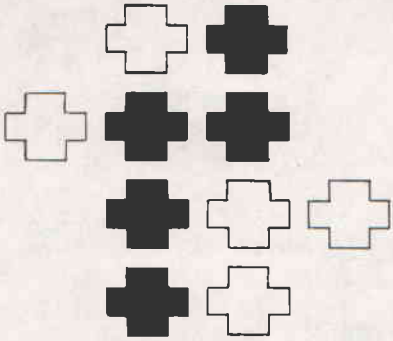


Figure 3. Apple® Flip-It.

```
1820 X = 54: GOSUB 1940:Y1 = Y1 +
     1: IF  SCRN( X1,Y1) = 6 THEN
     1930
1830 X = 55: GOSUB 1940:Y1 = Y1 +
     1: IF  SCRN( X1,Y1) = 6 THEN
     1930
1840 P2 = 0: FOR Y = 1 TO 8: FOR
     A = 1 TO 8
1850 X = 10 * Y + A: GOSUB 1940:Y
     1 = Y1 + 1:X1 = X1 + 1: IF  SCRN(
     X1,Y1) = 6 THEN 1870
1860 P2 = P2 + 1: IF  SCRN( X1,Y1
     ) = YOUR THEN SY = SY + 1
1870  NEXT A,Y: IF P2 - SY = 0 OR
     SY = 0 THEN 1900
1880  IF P2 = 64 THEN 1910
1890  COLOR=  SCRN( XX,YY - 1): GOTO
     220
1900  GOSUB 1920: HOME : PRINT "Y
     OU HAVE TO GIVE ME A CHANCE!
     ":P2 = 0:SY = 0: FOR X = 1 TO
     600: NEXT X: GOTO 1690
1910  GOSUB 1920: HOME : PRINT "Y
     OU DIDN'T LEAVE ANY ROOM!": FOR
     X = 1 TO 600: NEXT X: GOTO 1
     690
1920  FOR X = 1 TO 100: POKE  - 1
     6336, PEEK ( - 16336): NEXT
     X: RETURN
1930  HOME : GOSUB 1920: PRINT "T
     HE CENTRAL SQUARES MUST BE F
     ILLED": FOR X = 1 TO 1000: NEXT
     X: GOTO 1690
1940 X1 = 5 *  FN MT(X) - 5:Y1 =
      INT (X / 10) * 5 - 5: RETURN
1950 P2 = 4:SY = 2: GOTO 210
```

Print Instructions if requested.

```
1960  TEXT : HOME : VTAB 12: PRINT
     "DO YOU NEED INSTRUCTIONS? "
     ;
1970  GET A$: IF A$ = "N" THEN  RETURN

1980  IF A$ <  > "Y" THEN 1970
1990  HOME : VTAB 2: HTAB 15: PRINT
     "FLIP-IT 1" CHR$ (91): PRINT
     : PRINT "THE OBJECT OF THIS
     GAME IS TO COMPLETELY"
2000  PRINT "FILL THE BOARD WITH
     AS MANY PIECES OF": PRINT : PRINT
     "YOUR COLOR AS YOU CAN.  TO
     DO THIS YOU": PRINT
2010  PRINT "OUTFLANK YOUR OPPONE
     NT'S PIECES AND FLIP": PRINT
     "THEM TO YOUR COLOR.  OUTFLA
     NKING CAN": PRINT
2020  PRINT "OCCUR HORIZONTALLY,
     VERTICALLY, OR DIAG-": PRINT
     "ONALLY. THE GAME ENDS WHEN
     THE BOARD IS": PRINT
2030  PRINT "FULL OR A PLAYER CAN
     NOT MAKE A MOVE.": PRINT : PRINT
     "WHOEVER HAS THE MOST PIECES
      WINS."
2040  VTAB 24: PRINT "    (PRESS
     SPACE BAR TO CONTINUE)";
2050  IF  PEEK ( - 16384) < 128 THEN
     2050
2060  POKE  - 16368,0: HOME : VTAB
     2: PRINT "YOU MAKE YOUR MOVE
      BY TURNING PADDLE 0": PRINT

2070  PRINT "UNTIL THE CURSOR IS
     IN THE SQUARE YOU": PRINT : PRINT
     "WANT, AND THEN PRESSING THE
      BUTTON.": PRINT
2080  PRINT : PRINT "DURING PLAY
     YOU MAY ALSO CHOOSE ONE OF":
      PRINT : PRINT "THE FOLLOWIN
     G OPTIONS BY PRESSING THE": PRINT
     : PRINT "KEY INDICATED:"
2090  PRINT : PRINT "   Q - QUIT
     THE GAME": PRINT : PRINT "
      N - NOT ABLE TO MAKE A MOVE
     ": PRINT
2100  PRINT "   B - ASK THE COMPU
     TER TO RECOMMEND": PRINT "
          YOUR BEST MOVE.": PRINT

2110  VTAB 24: PRINT "    (PRESS
     SPACE BAR TO BEGIN)";
2120  IF  PEEK ( - 16384) < 128 THEN
     2120
2130  POKE  - 16368,0
2140 X = A(B): RETURN
```

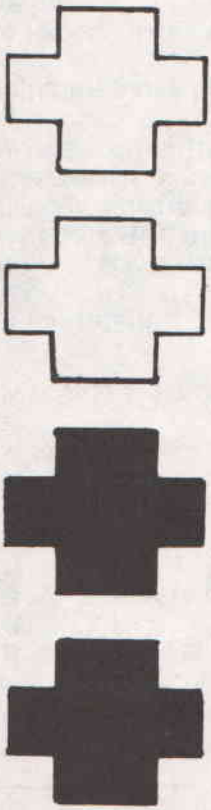
Data to set up square priority for the computer's strategy.

```
2150 A$ = "1181188831611383168638
     683363366641511484158548584 3
     533464356546564252247425754 7
     573262237326763767217 1": RETURN
```

```
2160 A$ = "12821787287822722777":
      RETURN
```

Chip-count update.

```
2170  POKE 34,22: HOME : PRINT "Y
      OUR CHIPS =";SY;: HTAB 20: PRINT
      "APPLE'S CHIPS =";P2 - SY: POKE
      34,23: RETURN
```



For APPLE® FLIP-IT II

LINES	SWAT CODE	LENGTH
10 - 110	NJ	545
120 - 230	XT	432
240 - 350	WK	275
360 - 470	UV	372
480 - 590	SJ	486
600 - 710	DM	429
720 - 830	QJ	372
840 - 950	UC	372
960 - 1070	WL	404
1080 - 1190	XH	495
1200 - 1310	MJ	413
1320 - 1430	QW	270
1440 - 1550	LL	264
1560 - 1670	PN	332
1680 - 1790	HS	433
1800 - 1910	SQ	461
1920 - 2020	DS	589
2030 - 2100	KP	555
2110 - 2170	XB	309

```
SS SS SS SS SS SS SS SS SS SS SS
SS                             SS
SS        TRS-80 BASIC         SS
SS        'FLIP-IT II'         SS
SS  AUTHOR: MICHAEL PRESCOTT   SS
SS    TRANSL: 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 available on #39 SoftSide CV and DV.

Initialization.

```
10 CLS:CLEAR200:DEFINTA-Z:DIMA(64),B(99):A$="1181188831611383168
6386833633666415114841585485833346435654656425224742575475732 6
2237326763767217112821787287822722777445555445":Z=1
20 FORX=1TO128STEP2:A(Z)=ASC(MID$(A$,X,1))-48+(ASC(MID$(A$,X+1,1
))-48)*10:Z=Z+1:NEXT:GOSUB580:CLS
```

Initialize game board and graphic strings.

```
30 FORX=0TO99:B(X)=0:NEXT:B(45)=-1:B(54)=-1:B(44)=1:B(55)=1:BP=2
:WP=2:P2=4
40 PDL=0:B=0:FL=0:P2=0:RF=0:BL=-1:WH=1:OP=0:YO=0:FORX=0TO9:B(X)=
9:B(X+90)=9:NEXT:FORX=9TO89STEP10:B(X)=9:B(X+1)=9:NEXT
50 B$=CHR$(191)+STRING$(3,131)+CHR$(191)+STRING$(5,24)+CHR$(26)+
STRING$(5,131):W$=STRING$(5,191)+STRING$(5,24)+CHR$(26)+STRING$(
5,131)
```

Get choice of color, who goes first, and whether or not a normal board is to be used. Also set up players' pieces.

```
60 CLS:F=0:A$="":PRINT@24,"F L I P - I T":PRINT:PRINT:INPUT"WHIC
H COLOR DO YOU WANT (W OR B) ";A$:IFLEFT$(A$,1)="B"THEN80
70 YO$=W$:OP$=B$:YO=WH:OP=BL:GOTO90
80 YO$=B$:OP$=W$:YO=BL:OP=WH
90 A$="":PRINT:INPUT"DO YOU WANT TO GO FIRST (Y OR N) ";A$:IFLEF
T$(A$,1)<>"N"THEN F=1
100 A$="":PRINT:INPUT"WANT TO SET UP YOUR OWN GAME (Y OR N) ";A$
:CLS:GOSUB110:IFLEFT$(A$,1)="Y"THENB(44)=0:B(45)=0:B(54)=0:B(55)
=0:WP=0:BP=0:P2=0:GOTO210ELSEIFF=1THEN290ELSEGOSUB120:GOTO360
```

Draw game board.

```
110 FORX=0TO7:FORY=0TO7:PRINT@X*128+Y*7,STR$(X+1);STR$(Y+1);:NEX
TY:NEXTX:RETURN
```

Plot pieces and update game display data.

```
120 P2=0:WP=0:BP=0:FORX=0TO99:IFB(X)<>0ANDB(X)<>9THENY=INT((X-11
)/10)*128+((X-11)-INT((X-11)/10)*10)*7 ELSE 150
130 IFB(X)=1THENPRINT@Y,W$;:WP=WP+1:P2=P2+1
140 IFB(X)=-1THENPRINT@Y,B$;:BP=BP+1:P2=P2+1
150 NEXT:FORY=0TO47:SET(110,Y):NEXT:PRINT@56,"FLIP-IT";:PRINT@18
5,"W =";WP;:PRINT@249,"B =";BP;:RETURN
```

No-possible-move command.

```
160 PRINT@376,"CHECKING";:FORZ=1TO60:PDL=A(Z):P=PDL:C=YO:SC=OP:F
L=0:GOSUB460:IFLG=1THEN170ELSENEXT:IFIP=1THEN390ELSEYP=1:GOTO360
170 PRINT@376,"?WHAT   ";:PRINT@440,"ABOUT  ";:A$=LEFT$(STR$(PDL
),2)+" "+RIGHT$(STR$(PDL),1):FORX=1TO15:PRINT@504,A$;:FORY=0TO99
:NEXT:PRINT@504,"       ";:FORY=0TO99:NEXT:NEXT:GOTO300
```

Find player's best move.

```
180 PRINT@376,"LOOKING ";:FORZ=1TO60:PDL=A(Z):P=PDL:C=YO:SC=OP:F
L=0:GOSUB460:IFLG=1THEN190ELSENEXT:IFIP=1THEN390ELSEYP=1:GOTO360
190 PRINT@440,"LOOKS   ";:PRINT@504,"GOOD!";:A$=LEFT$(STR$(PDL),
2)+" "+RIGHT$(STR$(PDL),1):FORX=1TO15:PRINT@376,A$;:FORY=0TO99:N
EXT:PRINT@376,"       ";:FORY=0TO99:NEXT:NEXT:PRINT@504,"     ";
:GOTO300
```

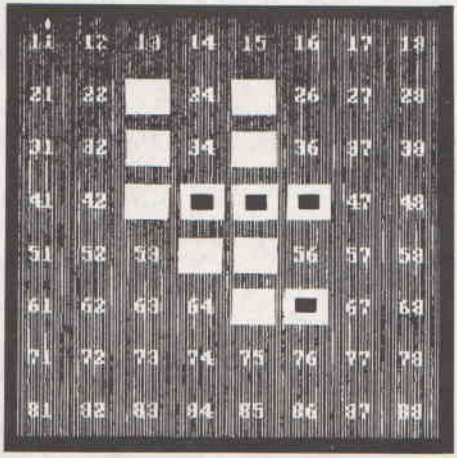


Figure 4. TRS-80® Flip-It screen simulation.

Quit-game routine.

```
200 A$="":CLS:PRINT@24,"F L I P - I T":PRINT:INPUT"DO YOU REALLY
 WANT TO QUIT (Y OR N) ";A$:IFLEFT$(A$,1)<>"Y"THENCLS:GOSUB110:G
OTO290ELSE400
```

Set up your own board.

```
210 RF=1:PRINT@504,"(D=DONE)";:GOSUB300:IFA$="D"THEN230ELSEPRINT
@376,"COLOR?  ";
220 A$=INKEY$:IFA$=""THEN220
230 IFA$="D"THENFORX=5TO7:PRINT@X*64+56,"        ";:NEXT:GOSUB11
0:P2=WH:WP=WH:BP=WH:IFF=1THEN290ELSEGOSUB120:GOTO360
240 PRINT@INT((PDL-11)/10)*128+((PDL-11)-INT((PDL-11)/10)*10)*7,
;
250 IFA$="W"THENPRINTW$;:B(PDL)=WH
260 IFA$="B"THENPRINTB$;:B(PDL)=BL
270 IFA$=CHR$(31)THENPRINTSTR$(INT(PDL/10));" ";RIGHT$(STR$(PDL)
,1);" ";STRING$(5,24);CHR$(26);"     ";:B(PDL)=0
280 GOTO210
```

Get player's move.

```
290 IFP2=64ORWP=0ORBP=0THEN390ELSEGOSUB120:IFP2=64ORWP=0ORBP=0TH
EN390
300 PDL=0:YP=0:PRINT@440,"        ";:PRINT@376,;:IFRF=1THENPRINT"
SQUARE? ";ELSEPRINT"MOVE?   ";
310 FORX=0TO1
320 A$=INKEY$:IFA$=""THEN320ELSEIFA$="C"ANDRF=0THENF=1:GOTO210EL
SEIFA$="R"ANDRF=0THENCLS:GOSUB110:GOSUB120:GOTO300ELSEIFA$="Q"AN
DRF=0THEN200ELSEIFA$="N"ANDRF=0THEN160ELSEIFA$="B"ANDRF=0THEN180
ELSEIFA$="D"ANDRF=1THENRF=0:RETURN
330 IFA$<"1"ORA$>"8"THENPRINT@440,"INVALID";:FORY=1TO555:NEXT:GO
TO300
340 PRINT@441+X*2,A$;:PDL=PDL+VAL(A$)*10[(1-X):NEXT:IFRF=1THENRF
=0:RETURN
350 C=YO:SC=OP:FL=1:GOSUB460:IFLG=0THENA$="0":GOTO330ELSEGOSUB12
0
```

Choose computer's move.

```
360 IFP2=64ORWP=0ORBP=0THEN390
370 IP=0:PRINT@376,"MAYBE   ";:FORZ=1TO64:PDL=A(Z):PRINT@440,LEF
T$(STR$(PDL),2)" "RIGHT$(STR$(PDL),1);:P=PDL:IF B(P)<>0THENNEXTZ
ELSEC=OP:SC=YO:FL=1:GOSUB460:IFLG=0THENNEXTZELSE290
```

Computer must pass.

```
380 IF P2<>64 AND YP=0 THEN PRINT@569,"PASS!";:IP=1:FORX=0TO555:
NEXTX:PRINT@569,"     ";:GOTO290
```

End-game routine.

```
390 PRINT@568,"NO MOVES";:PRINT@634,"LEFT!";:PRINT@760,"$ GAME $
";:PRINT@824,"$ OVER $";:FORX=0TO5000:IFINKEY$<>" "THENNEXTX
400 CLS:PRINT@24,"F L I P - I T":PRINT:PRINT"FINAL SCORE:":PRINT
:PRINT"    WHITE ="WP:PRINT"    BLACK ="BP:PRINT
410 IFWP=BPTHENPRINT"CONGRATULATIONS. IT'S A DRAW.":PRINT:PRINT"
WE APPEAR TO BE EVENLY MATCHED.":PRINT:GOTO440ELSEIF(WP>BPANDOP=
WH)OR(BP>WPANDOP=BL)THENPRINT"THANK YOU FOR A STIMULATING GAME":
PRINT:PRINT"I HAVE WON THIS GAME BY";:GOTO430
420 PRINT"CONGRATULATIONS!":PRINT:PRINT"YOU HAVE WON THIS GAME B
Y";
430 PRINTABS(WP-BP)"CHIPS.":PRINT
```

Check for new game.

```
440 A$="":INPUT"WOULD YOU LIKE TO PLAY AGAIN (Y OR N) ";A$:IFLEF
T$(A$,1)<>"N"THENCLS:GOTO 30
450 CLEAR50:PRINT:PRINT"GOOD BYE...":END
```

Legal check routine. Combines a search for legal moves with a routine to flip pieces.

```
460 LG=0::IFB(PDL)<>0THEN570
470 FORY=1TO8:F=0:ONYGOTO480,490,500,510,520,530,540,550
480 P=PDL:FORX=1TO8:P=P+10:IFP>99THEN560ELSEIFB(P)=0OR(B(P)=CAND
X=1)ORB(P)=9THEN560ELSEIFB(P)=SCTHENIFF=1THENB(P)=C:NEXTX:GOTO56
0ELSENEXTX:GOTO560ELSEF=1:LG=1:P=PDL:IFFL=0THEN570ELSEB(P)=C:GOT
O480
490 P=PDL:FORX=1TO8:P=P-10:IFP<0THEN560ELSEIFB(P)=0OR(B(P)=CANDX
=1)ORB(P)=9THEN560ELSEIFB(P)=SCTHENIFF=1THENB(P)=C:NEXTX:GOTO560
ELSENEXTX:GOTO560ELSEF=1:LG=1:P=PDL:IFFL=0THEN570ELSEB(P)=C:GOTO
490
500 P=PDL:FORX=1TO8:P=P+1:IFP>99THEN560ELSEIFB(P)=0OR(B(P)=CANDX
=1)ORB(P)=9THEN560ELSEIFB(P)=SCTHENIFF=1THENB(P)=C:NEXTX:GOTO560
ELSENEXTX:GOTO560ELSEF=1:LG=1:P=PDL:IFFL=0THEN570ELSEB(P)=C:GOTO
500
510 P=PDL:FORX=1TO8:P=P-1:IFP<0THEN560ELSEIFB(P)=0OR(B(P)=CANDX=
1)ORB(P)=9THEN560ELSEIFB(P)=SCTHENIFF=1THENB(P)=C:NEXTX:GOTO560E
LSENEXTX:GOTO560ELSEF=1:LG=1:P=PDL:IFFL=0THEN570ELSEB(P)=C:GOTO5
10
520 P=PDL::FORX=1TO8:P=P+9:IFP>99THEN560ELSEIFB(P)=0OR(B(P)=CAND
X=1)ORB(P)=9THEN560ELSEIFB(P)=SCTHENIFF=1THENB(P)=C:NEXTX:GOTO56
0ELSENEXTX:GOTO560ELSEF=1:LG=1:P=PDL:IFFL=0THEN570ELSEB(P)=C:GOT
O520
530 P=PDL:FORX=1TO8:P=P-9:IFP<0THEN560ELSEIFB(P)=0OR(B(P)=CANDX=
1)ORB(P)=9THEN560ELSEIFB(P)=SCTHENIFF=1THENB(P)=C:NEXTX:GOTO560E
LSENEXTX:GOTO560ELSEF=1:LG=1:P=PDL:IFFL=0THEN570ELSEB(P)=C:GOTO5
30
```

```
540 P=PDL:FORX=1TO8:P=P+11:IFP>99THEN560ELSEIFB(P)=0OR(B(P)=CAND
X=1)ORB(P)=9THEN560ELSEIFB(P)=SCTHENIFF=1THENB(P)=C:NEXTX:GOTO56
0ELSENEXTX:GOTO560ELSEF=1:LG=1:P=PDL:IFFL=0THEN570ELSEB(P)=C:GOT
O540
550 P=PDL:FORX=1TO8:P=P-11:IFP<0THEN560ELSEIFB(P)=0OR(B(P)=CANDX
=1)ORB(P)=9THEN560ELSEIFB(P)=SCTHENIFF=1THENB(P)=C:NEXTX:GOTO560
ELSENEXTX:GOTO560ELSEF=1:LG=1:P=PDL:IFFL=0THEN570ELSEB(P)=C:GOTO
550
560 FORX=0TO1:NEXTX:NEXTY
570 FORX=0TO1:NEXTX:FORY=0TO1:NEXTY:RETURN
```

Game instructions.

```
580 A$="":PRINT@512,;:INPUT"NEED INSTRUCTIONS (Y OR N) ";A$:IFLE
FT$(A$,1)<>"Y"THENRETURN
590 CLS:PRINT@24,"F L I P - I T":PRINT:PRINT"THE OBJECT OF THIS
GAME IS TO COMPLETELY FILL THE BOARD WITH AS MANY PIECES OF YOUR
 COLOR AS YOU CAN. TO DO THIS YOU MUST OUT-  FLANK YOUR OPPONENT
'S PIECES AND FLIP THEM TO YOUR COLOR."
600 PRINT:PRINT"OUTFLANKING CAN OCCUR HORIZONTALLY, VERTICALLY,
OR DIAGONALLY.  THE GAME ENDS WHEN THE BOARD IS FULL, OR WHEN BO
TH PLAYERS CAN'TMOVE. AT THAT TIME, WHOEVER HAS THE MOST PIECES
WINS."
610 PRINT@977,"(PRESS <ENTER> TO CONTINUE)";
620 IFINKEY$<>CHR$(13)THEN620
630 CLS:PRINT@24,"F L I P - I T":PRINT:PRINT"YOU MAKE MOVES BY E
NTERING THE ROW NUMBER COMBINED WITH THE     COLUMN NUMBER. FOR
EXAMPLE: THE UPPER RIGHT-HAND CORNER IS THE  POSITION 18 - THAT
IS 1 FOR THE FIRST ROW AND 8 FOR THE EIGHTH"
640 PRINT"COLUMN. IF YOU MAKE A MISTAKE ENTERING A NUMBER, TYPE
A NINE FORANY OF 1HE NUMBERS AND IT WILL MARK THE ENTRY AS INVAL
ID.":PRINT"DURING THE GAME YOU ALSO HAVE THE FOLLOWING OPTIONS:"
:PRINT
650 PRINT"Q - TO QUIT THE GAME":PRINT"R - REDRAW THE GAME BOARD"
:PRINT"N - NO AVAILABLE MOVE":PRINT"C - CHANGE GAME BOARD SETUP"
:PRINT"B - ASK FOR THE BEST MOVE":PRINT@977,"(PRESS <ENTER> TO S
TART GAME)";
660 IFINKEY$<>CHR$(13)THEN660ELSERETURN
```

SWAT TABLE

For TRS-80® FLIP-IT II

LINES	SWAT CODE	LENGTH	LINES	SWAT CODE	LENGTH
10 - 50	UX	504	440 - 500	CG	611
60 - 120	0G	532	510 - 540	TE	576
130 - 190	OJ	625	550 - 600	YG	664
200 - 290	CQ	531	610 - 650	SG	650
300 - 370	ET	547	660 - 660	BF	21
380 - 430	SZ	520			

```
SS SS SS SS SS SS SS SS SS SS SS
SS                              SS
SS          ATARI  BASIC        SS
SS          'FLIP-IT II'        SS
SS  AUTHOR: MICHAEL PRESCOTT    SS
SS     TRANSL: 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 available on #39 SoftSide CV and DV.

Initialization.

```
10 N0=0:N1=1:N2=2:N3=3:N4=4:N5=5:N7=7:
N8=8:N10=10:N12=12:CLS=125:RESET=255:C
RS=752:KEY=764:SPK=53279
100 GRAPHICS N0:POKE CRS,RESET:? :DIM
A(60),A$(120):Z=N1:GOSUB 6020:FOR X=N1
 TO LEN(A$) STEP 2
110 FL=N0:A(Z)=ASC(A$(X,X))-48+(ASC(A$
(X+N1,X+N1))-48)*N10:Z=Z+N1:NEXT X:CLO
SE #N1:OPEN #N1,4,0,"K":GOSUB 5000
120 PDL=N0:B=N0:FL=N0:B=C=D:P2=N4:SY=N
2:PDL=N0:B=N0:F=N0
```

Introduction.

```
150 GRAPHICS N5:SETCOLOR N2,N0,N0:SETC
OLOR N4,N0,N0:SETCOLOR N0,N0,N0:COLOR
N1:POKE CRS,RESET:?
160 FOR X=N12 TO 39 STEP 0.5:SOUND N0,
40,N4,(X-N12)/N3:PLOT X,N0:DRAWTO X,38
:PLOT 78-X,N0:DRAWTO 78-X,38:NEXT X
170 COLOR N0:FOR X=N1 TO N7:PLOT N7*X+
11,N0:DRAWTO N7*X+11,39:PLOT N12,N5*X-
N1:DRAWTO 67,N5*X-N1:NEXT X
180 FOR X=N0 TO N7:FOR Y=N1 TO N12:SET
COLOR N0,Y,N4:A=(X*N12)+Y*N12:SOUND N0
,A,N10,N4:SOUND N1,A+1,N10,N4:NEXT Y
190 NEXT X:SOUND N0,N0,N0,N0:SOUND N1,
N0,N0,N0
```

Get player's choice of color.

```
200 SETCOLOR N1,N0,N10:? CHR$(CLS);"WH
ICH COLOR? (W/B)";:GET #N1,A:IF A<>87
AND A<>66 THEN 200
205 IF A=87 THEN YOUR=N2:OPPONENT=N3:G
OTO 230
210 IF A=66 THEN YOUR=N3:OPPONENT=N2
230 XX=14:YY=N2:COLR=N1
```

Prompt for type of game and choose who goes first.

```
240 ? CHR$(CLS);"WANT TO GO FIRST? (Y/
N)";:GET #N1,A:F=N0:IF A=78 THEN 250
245 F=N1:IF A<>89 THEN 240
250 ? CHR$(CLS);"WANT TO SET UP YOUR O
WN GAME? (Y/N)";:GET #N1,A:IF A<>78 AN
D A<>89 THEN 250
255 IF A=89 THEN GOSUB 6010:GOTO 4000
260 TRAP 33333:GOSUB 6000:GOTO 600
300 ST=STICK(N0)
310 IF ST=15 THEN ST=N0
320 IF ST=14 AND PDL>N7 THEN PDL=PDL-N
8
330 IF ST=13 AND PDL<56 THEN PDL=PDL+N
8
340 IF ST=11 AND PDL>N0 THEN PDL=PDL-N
1
350 IF ST=N10 AND PDL>8 THEN PDL=PDL-9
360 IF ST=9 AND PDL<56 THEN PDL=PDL+N7
370 IF ST=N7 AND PDL<63 THEN PDL=PDL+N
1
380 IF ST=6 AND PDL>N7 THEN PDL=PDL-N7
390 IF ST=N5 AND PDL<55 THEN PDL=PDL+9
400 RETURN
```

Set up new game.

```
600 COLOR N3:X=45:GOSUB 1160:X=54:GOSU
B 1160:COLOR N2:X=44:GOSUB 1160:X=55:G
OSUB 1160
610 POKE KEY,RESET
620 IF F=N0 THEN 840
```

Update bar chart of players' pieces, get player's command, and update the square select cursor.

```
630 GOSUB 7000
640 IF PEEK(KEY)=RESET THEN 800
645 CO=PEEK(KEY):POKE KEY,RESET:IF CO<
>47 AND CO<>35 AND CO<>21 THEN 800
650 IF CO=47 THEN 2012
652 IF CO=35 THEN 680
654 GOSUB 700:GOTO 800
```

No-possible-move command.

```
680 GOSUB 700:IF YP=N1 THEN 840
682 SOUND N0,255,N12,15:FOR X=N1 TO 15
0:NEXT X:SOUND N0,N0,N0,N0:GOTO 800
```

Find the best possible move.

```
700 FL=N1:Q=YOUR:YOUR=OPPONENT:OPPONEN
T=Q:GOSUB 2000:IF YP=N1 THEN 710
702 FOR XXX=N1 TO 20:IF (XXX-INT(XXX/N
2)*N2)=N0 THEN COLOR N0
704 IF (XXX-INT(XXX/N2)*N2)<>N0 THEN C
OLOR N1
706 IF (XXX-INT(XXX/N5)*N5)<>N0 THEN G
OSUB 1160
708 NEXT XXX
710 Q=YOUR:YOUR=OPPONENT:OPPONENT=Q:FL
=N0:RETURN
```

Update and display player's move.

```
800 IF SY=N0 OR P2-SY=N0 OR P2=64 THEN
 2010
802 GOSUB 300:IF A=PDL THEN 810
804 A=PDL:LOCATE XX,YY-N1,AZ:COLOR AZ:
PLOT XX,YY:XX=N7*(A-INT(A/N8)*N8)+14:Y
Y=INT(A/8)*N5+N2:LOCATE XX,YY-N1,COLR
806 COLOR N0:PLOT XX,YY
810 IF STRIG(N0)=N1 THEN 640
812 IF COLR<>N1 THEN 640
820 P3=SY:COLOR YOUR:P4=N0:D=N0:FOR XX
X=N1 TO N8:X=(A-INT(A/N8)*N8)+N1+N10*(
INT(A/N8)+N1):P1=N0:GOSUB 980+XXX*20
822 P4=P4+P1:IF P1<>N0 THEN D=D+N1
830 NEXT XXX:IF D=N0 THEN 640
832 SY=SY+P4-D+N1:P2=P2+N1
```

Find computer's best move and display it.

```
840 IF SY=N0 OR P2-SY=N0 OR P2=64 THEN
 2010
842 GOSUB 7000:GOSUB 2000:FOR XXX=N1 T
O 42:IF (XXX-INT(XXX/N2)*N2)=N0 THEN C
OLOR N0
844 IF (XXX-INT(XXX/N2)*N2)<>N0 THEN C
OLOR N1
846 IF (XXX-INT(XXX/N3)*N3)=N0 THEN GO
SUB 1160
848 NEXT XXX:COLOR OPPONENT
900 P2=P2+N1:Q=YOUR:YOUR=OPPONENT:OPPO
NENT=Q:P4=N0:D=N0:FOR XXX=N1 TO N8:X=A
(B):P1=N0:GOSUB 980+20*XXX
905 IF P1<>N0 THEN D=D+N1:P4=P4+P1
910 NEXT XXX:SY=SY-P4+D
920 Q=YOUR:YOUR=OPPONENT:OPPONENT=Q:LO
CATE XX,YY-N1,COLR:GOTO 630
```

Search and capture pieces routines. Spaced every twenty lines.

```
1000 X1=N7*(X-INT(X/N10)*N10)+N7:Y1=IN
T(X/N10)*N5-N3:FOR X2=N1 TO N7:Y1=Y1-N
5:IF Y1<N0 THEN RETURN
1005 LOCATE X1,Y1,Z:IF Z=YOUR AND X2=N
1 OR Z=N1 THEN RETURN
1010 IF Z=OPPONENT THEN NEXT X2
1012 IF X2=N8 THEN RETURN
1014 GOSUB 1160:P1=X2:FOR X3=N1 TO X2:
X=X-N10:GOSUB 1160:NEXT X3:RETURN
1020 X1=N7*(X-INT(X/N10)*N10)+N7:Y1=IN
T(X/N10)*N5-N3:FOR X2=N1 TO N7:Y1=Y1+N
5:IF Y1>38 THEN RETURN
1025 LOCATE X1,Y1,Z:IF Z=YOUR AND X2=N
1 OR Z=N1 THEN RETURN
1030 IF Z=OPPONENT THEN NEXT X2
1032 IF X2=N8 THEN RETURN
1034 GOSUB 1160:P1=X2:FOR X3=N1 TO X2:
X=X+N10:GOSUB 1160:NEXT X3:RETURN
1040 X1=N7*(X-INT(X/N10)*N10)+N7:Y1=IN
T(X/N10)*N5-N3:FOR X2=N1 TO N7:X1=X1-N
7:IF X1<N12 THEN RETURN
1045 LOCATE X1,Y1,Z:IF Z=YOUR AND X2=N
1 OR Z=N1 THEN RETURN
1050 IF Z=OPPONENT THEN NEXT X2
1052 IF X2=N8 THEN RETURN
```

```
1054 GOSUB 1160:P1=X2:FOR X3=N1 TO X2:
X=X-N1:GOSUB 1160:NEXT X3:RETURN
1060 X1=N7*(X-INT(X/N10)*N10)+N7:Y1=IN
T(X/N10)*N5-N3:FOR X2=N1 TO N7:X1=X1+N
7:IF X1>66 THEN RETURN
1065 LOCATE X1,Y1,Z:IF Z=YOUR AND X2=N
1 OR Z=N1 THEN RETURN
1070 IF Z=OPPONENT THEN NEXT X2
1072 IF X2=N8 THEN RETURN
1074 GOSUB 1160:P1=X2:FOR X3=N1 TO X2:
X=X+N1:GOSUB 1160:NEXT X3:RETURN
1080 X1=N7*(X-INT(X/N10)*N10)+N7:Y1=IN
T(X/N10)*N5-N3:FOR X2=N1 TO N7:X1=X1-N
7:Y1=Y1-N5
1082 IF X1<N12 OR Y1<N0 THEN RETURN
1085 LOCATE X1,Y1,Z:IF Z=YOUR AND X2=N
1 OR Z=N1 THEN RETURN
1090 IF Z=OPPONENT THEN NEXT X2
1092 IF X2=N8 THEN RETURN
1094 GOSUB 1160:P1=X2:FOR X3=N1 TO X2:
X=X-11:GOSUB 1160:NEXT X3:RETURN
1100 X1=N7*(X-INT(X/N10)*N10)+N7:Y1=IN
T(X/N10)*N5-N3:FOR X2=N1 TO N7:X1=X1+N
7:Y1=Y1-N5
1102 IF X1>66 OR Y1<N0 THEN RETURN
1105 LOCATE X1,Y1,Z:IF Z=YOUR AND X2=N
1 OR Z=N1 THEN RETURN
1110 IF Z=OPPONENT THEN NEXT X2
1112 IF X2=N8 THEN RETURN
1114 GOSUB 1160:P1=X2:FOR X3=N1 TO X2:
X=X-9:GOSUB 1160:NEXT X3:RETURN
1120 X1=N7*(X-INT(X/N10)*N10)+N7:Y1=IN
T(X/N10)*N5-N3:FOR X2=N1 TO N7:X1=X1+N
7:Y1=Y1+N5
1122 IF X1>66 OR Y1>38 THEN RETURN
1125 LOCATE X1,Y1,Z:IF Z=YOUR AND X2=N
1 OR Z=N1 THEN RETURN
1130 IF Z=OPPONENT THEN NEXT X2
1132 IF X2=N8 THEN RETURN
1134 GOSUB 1160:P1=X2:FOR X3=N1 TO X2:
X=X+11:GOSUB 1160:NEXT X3:RETURN
1140 X1=N7*(X-INT(X/N10)*N10)+N7:Y1=IN
T(X/N10)*N5-N3:FOR X2=N1 TO N7:X1=X1-N
7:Y1=Y1+N5
1142 IF X1<N12 OR Y1>38 THEN RETURN
1145 LOCATE X1,Y1,Z:IF Z=YOUR AND X2=N
1 OR Z=N1 THEN RETURN
1150 IF Z=OPPONENT THEN NEXT X2
1152 IF X2=N8 THEN RETURN
1154 GOSUB 1160:P1=X2:FOR X3=N1 TO X2:
X=X+9:GOSUB 1160:NEXT X3:RETURN
```

Plot chip at selected location.

```
1160 X1=N7*(X-INT(X/N10)*N10)+N5:Y1=IN
T(X/N10)*N5-N5:FOR X1=X1 TO X1+N5:PLOT
 X1,Y1:DRAWTO X1,Y1+N3:NEXT X1
```

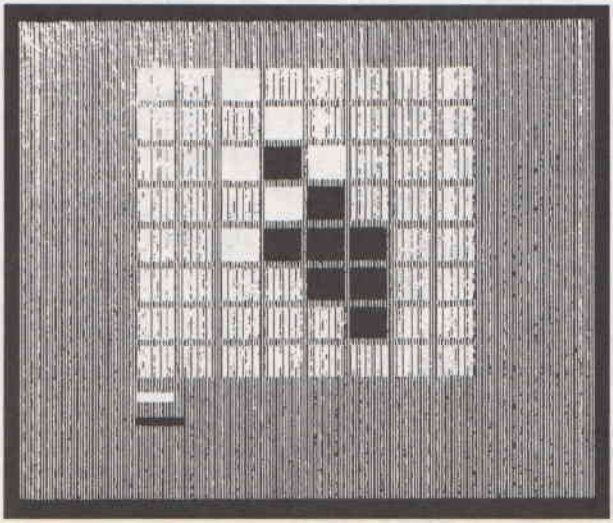


Figure 5. Atari® Flip-It screen.

```
1165 FOR Z=N1 TO 6:POKE SPK,N0:NEXT Z:
RETURN
```

Search through the best moves for the computer. If none are found, pass or concede the game.

```
2000 POKE 77,N0:IF P2-SY=N0 OR SY=N0 T
HEN 2010
2001 IF FL=N0 THEN IP=N0
2002 IF FL=N1 THEN YP=N0
2003 FOR B=N1 TO 60:X=A(B):X1=N7*(X-IN
T(X/N10)*N10)+N7:Y1=INT(X/N10)*N5-N3:P
OKE SPK,N0:POKE SPK,N0:LOCATE X1,Y1,Z
2004 IF Z=N1 THEN GOSUB 3000
2005 NEXT B:IF P2-SY<>N0 AND SY<>N0 AN
D FL=N0 AND YP=N0 THEN 2008
2006 IF P2-SY<>N0 AND SY<>N0 AND FL=N1
 AND IP=N0 THEN YP=N1:RETURN
2007 GOTO 2010
2008 IP=N1:FOR ST=170 TO 220:SOUND N0,
ST,N10,N10:SOUND N1,ST+1,N10,N10:POKE
SPK,N0:POKE SPK,N0:NEXT ST
2009 SOUND N0,N0,N0,N0:SOUND N1,N0,N0,
N0:GOTO 630
2010 FOR X=N0 TO 200:SOUND N0,X,N12,6:
NEXT X:SOUND N0,N0,N0,N0:GRAPHICS N0
2012 POKE CRS,RESET:? "NO LEGAL MOVES
LEFT!":? :IF P2-SY>SY THEN 2030
2014 IF P2-SY=SY THEN 2040
2020 ? "YOU HAVE WON BY ";N2*SY-P2;" C
HIPS.":? :? "CONGRATULATIONS!":GOTO 20
50
2030 ? "I HAVE WON BY ";P2-N2*SY;" CHI
PS.":? :? "THANK YOU FOR A STIMULATING
 GAME.":GOTO 2050
2040 ? "CONGRATULATIONS. IT'S A DRAW."
:? :? "WE APPEAR TO BE EVENLY MATCHED.
"
2050 FOR X=N1 TO N12:POKE SPK,N0:POKE
SPK,N0:NEXT X:? :? "WOULD YOU LIKE TO
PLAY AGAIN?":POKE KEY,RESET
2060 CO=PEEK(KEY):IF CO=43 THEN POKE K
EY,RESET:GOTO 120
2062 IF CO=35 THEN POKE KEY,RESET:POKE
 CRS,N0:? :? "GOOD BYE...":? :END
2064 GOTO 2060
```

The computer has found an empty square; now check to see if it is legal to put a chip there and flip pieces.

```
3000 FOR C=N1 TO N8:X1=N7*(X-INT(X/N10
)*N10)+6:Y1=INT(X/N10)*N5-N3:POKE SPK,
N0:POKE SPK,N0:GOSUB 2990+20*C
3005 NEXT C:RETURN
3010 FOR D=N1 TO N7:Y1=Y1+N5:IF Y1>38
THEN RETURN
3012 LOCATE X1,Y1,Z:IF Z=N1 OR Z=OPPON
ENT AND D=N1 THEN RETURN
3014 IF Z=YOUR THEN NEXT D
3020 IF D=N8 THEN RETURN
3022 POP :POP :POP :POP :POP :GOTO 515
0
3030 FOR D=N1 TO N7:Y1=Y1-N5:IF Y1<N0
THEN RETURN
3032 LOCATE X1,Y1,Z:IF Z=N1 OR Z=OPPON
ENT AND D=N1 THEN RETURN
3034 IF Z=YOUR THEN NEXT D
3040 IF D=N8 THEN RETURN
3042 POP :POP :POP :POP :POP :GOTO 515
0
3050 FOR D=N1 TO N7:X1=X1-N7:IF X1<N12
 THEN RETURN
3052 LOCATE X1,Y1,Z:IF Z=N1 OR Z=OPPON
ENT AND D=N1 THEN RETURN
3054 IF Z=YOUR THEN NEXT D
3060 IF D=N8 THEN RETURN
3062 POP :POP :POP :POP :POP :GOTO 515
0
3070 FOR D=N1 TO N7:X1=X1+N7:IF X1>66
THEN RETURN
3072 LOCATE X1,Y1,Z:IF Z=N1 OR Z=OPPON
ENT AND D=N1 THEN RETURN
3074 IF Z=YOUR THEN NEXT D
3080 IF D=N8 THEN RETURN
3082 POP :POP :POP :POP :POP :GOTO 515
0
3090 FOR D=N1 TO N7:X1=X1+N7:Y1=Y1+N5:
IF X1>66 OR Y1>38 THEN RETURN
3092 LOCATE X1,Y1,Z:IF Z=N1 OR Z=OPPON
ENT AND D=N1 THEN RETURN
3094 IF Z=YOUR THEN NEXT D
3100 IF D=N8 THEN RETURN
3102 POP :POP :POP :POP :POP :GOTO 515
0
```

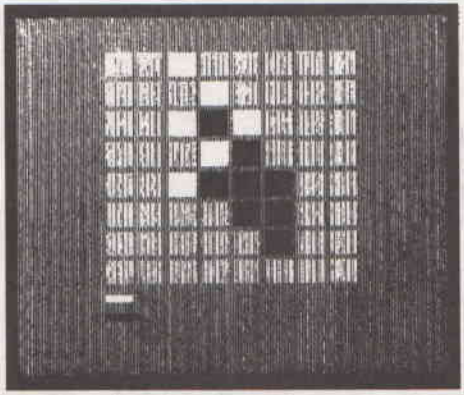


Figure 6. Atari® Flip-It screen.

```
3110 FOR D=N1 TO N7:X1=X1-N7:Y1=Y1+N5:
IF X1<N12 OR Y1>38 THEN RETURN
3112 LOCATE X1,Y1,Z:IF Z=N1 OR Z=OPPON
ENT AND D=N1 THEN RETURN
3114 IF Z=YOUR THEN NEXT D
3120 IF D=N8 THEN RETURN
3122 POP :POP :POP :POP :POP :GOTO 515
0
3130 FOR D=N1 TO N7:X1=X1-N7:Y1=Y1-N5:
IF X1<N12 OR Y1<N0 THEN RETURN
3132 LOCATE X1,Y1,Z:IF Z=N1 OR Z=OPPON
ENT AND D=N1 THEN RETURN
3134 IF Z=YOUR THEN NEXT D
3140 IF D=N8 THEN RETURN
3142 POP :POP :POP :POP :POP :GOTO 515
0
3150 FOR D=N1 TO N7:X1=X1+N7:Y1=Y1-N5:
IF X1>66 OR Y1<N0 THEN RETURN
3152 LOCATE X1,Y1,Z:IF Z=N1 OR Z=OPPON
ENT AND D=N1 THEN RETURN
3154 IF Z=YOUR THEN NEXT D
3160 IF D=N8 THEN RETURN
3162 POP :POP :POP :POP :POP :GOTO 515
0
```

Set up your own game.

```
4000 TRAP 33333:POKE CRS,RESET:? CHR$(
CLS);"SET UP? (1=REGULAR, 2=MANUAL)";:
P2=N0:SY=N0
4010 GET #N1,CO:POKE KEY,RESET:CO=CO-4
8:IF CO<N1 OR CO>N2 THEN 4010
4020 IF CO=N1 THEN 4250
4050 ? CHR$(CLS);"PRESS ESC WHEN FINIS
HED,":? "USE BUTTON TO SELECT SQUARE"
4060 GOSUB 300:A=PDL:LOCATE XX,YY-N1,A
Z:COLOR AZ:PLOT XX,YY:XX=N7*(A-INT(A/N
8)*N8)+14:YY=INT(A/8)*N5+N2
4062 LOCATE XX,YY-N1,COLR:COLOR N0:PLO
T XX,YY
4070 IF PEEK(KEY)=RESET THEN 4080
4072 CO=PEEK(KEY):POKE KEY,RESET:IF CO
=28 THEN LOCATE XX,YY-N1,AZ:COLOR AZ:P
LOT XX,YY:GOTO 4160
4080 IF STRIG(N0)=N1 THEN 4060
4085 ? CHR$(CLS);
4090 B=N0:Z=N0:? "WHICH COLOR? (W/B/G)
";:GET #N1,ST:IF ST<>87 AND ST<>66 AND
 ST<>71 THEN 4085
4095 X=((A-INT(A/N8)*N8)+N1)+(N10*(INT
(A/N8)+N1)):GOSUB 4240:LOCATE X1,Y1-N1
,AZ:COLOR N3:IF ST=87 THEN COLOR N2
4100 IF ST=71 THEN COLOR N1:IF AZ=N1 T
HEN 4050
4135 IF ST=71 THEN P2=P2-N1
4140 GOSUB 1160:IF AZ=N1 THEN P2=P2+N1
4145 GOTO 4050
4150 IF P2=N0 THEN 4140
4155 GOSUB 4220:? "YOU'RE MAKING UP TH
IS GAME!":? "A PIECE COULDN'T GET THER
E!":FOR X=N1 TO 300:NEXT X:GOTO 4050
4160 X=44:GOSUB 4240:Y1=Y1+N1:LOCATE X
1,Y1,Z:IF Z=N1 THEN 4230
4162 X=45:GOSUB 4240:Y1=Y1+N1:LOCATE X
1,Y1,Z:IF Z=N1 THEN 4230
```

```
4170 X=54:GOSUB 4240:Y1=Y1+N1:LOCATE X
1,Y1,Z:IF Z=N1 THEN 4230
4172 X=55:GOSUB 4240:Y1=Y1+N1:LOCATE X
1,Y1,Z:IF Z=N1 THEN 4230
4174 P2=N0:FOR Y=N1 TO N8:FOR A=N1 TO
N8
4180 X=N10*Y+A:GOSUB 4240:Y1=Y1+N1:X1=
X1+N1:LOCATE X1,Y1,Z:IF Z=N1 THEN 4190
4182 P2=P2+N1:IF Z=YOUR THEN SY=SY+N1
4190 NEXT A:NEXT Y:IF P2-SY=N0 OR SY=N
0 THEN 4200
4192 IF P2=64 THEN 4210
4194 GOSUB 6000:GOTO 610
4200 GOSUB 4220:? CHR$(CLS);"YOU HAVE
TO GIVE ME A CHANCE!":P2=N0:SY=N0:FOR
X=N1 TO 300:NEXT X:GOTO 4050
4210 GOSUB 4220:? CHR$(CLS);"YOU DIDN'
T LEAVE ANY ROOM!":FOR X=N1 TO 300:NEX
T X:GOTO 4050
4220 ? CHR$(CLS);:FOR X=N1 TO N12:POKE
 SPK,N0:POKE SPK,N0:NEXT X:RETURN
4230 GOSUB 4220:? CHR$(CLS);"THE CENTR
AL SQUARES MUST BE FILLED!":FOR X=N1 T
O 300:NEXT X:GOTO 4050
4240 X1=N7*(X-INT(X/N10)*N10)+N7:Y1=IN
T(X/N10)*N5-N3:RETURN
4250 GOSUB 6000:P2=N4:SY=N2:GOTO 600
```

Print instructions for game.

```
5000 POKE CRS,RESET:POSITION N2,N12:?
"NEED INSTRUCTIONS? (Y/N)":GET #N1,A:I
F A=78 THEN RETURN
5005 IF A<>89 THEN 5000
5010 ? CHR$(CLS);:POSITION 14,N2:? " F
LIP-IT ":? :? "    THE OBJECT OF THIS
GAME IS TO"
5020 ? "COMPLETELY FILL THE BOARD WITH
 AS":? "MANY PIECES OF YOUR COLOR AS Y
OU CAN."
5030 ? "TO DO THIS YOU MUST OUTFLANK Y
OUR":? "OPPONENT'S PIECES AND FLIP THE
M TO"
5040 ? "YOUR COLOR. OUTFLANKING CAN OC
CUR":? "HORIZONTALLY, VERTICALLY, OR D
IAG-"
5050 ? "ONOLLY. THE GAME ENDS WHEN THE
 BOARD":? "IS FULL OR WHEN BOTH PLAYER
S CAN'T":? "MOVE. ":?
5060 ? "WHOEVER HAS THE MOST PIECES WI
NS.":POSITION N5,23:? "(PRESS ANY KEY
TO CONTINUE)";
5065 IF PEEK(KEY)=RESET THEN 5065
5070 POKE KEY,RESET:GRAPHICS N0:POKE C
RS,N1:POSITION N2,N2:? "YOU MAKE MOVES
 BY MOVING THE CURSOR"
5080 ? "WITH THE JOYSTICK TO THE SQUAR
E YOU":? "DESIRE AND PRESS FIRE":?
5090 ? "DURING THE GAME YOUR CAN ALSO
CHOOSE":? "ONE OF THE OPTIONS BY PRESS
ING THE":? "KEY INDICATED:":?
5100 ? "Q - TO QUIT THE GAME":? :? "N
- NOT ABLE TO MOVE":? :? "B - ASK FOR
THE BEST MOVE"
5110 POSITION N7,23:? "(PRESS ANY KEY
TO BEGIN)";
5130 IF PEEK(KEY)=RESET THEN 5130
5140 POKE KEY,RESET
5150 X=A(B):RETURN
```

Set up game colors and graphics mode.

```
6000 GRAPHICS N5+48
6010 TRAP 33333:SETCOLOR N0,N12,B:SETC
OLOR N1,N0,N10:SETCOLOR N2,N0,N0:SETCO
LOR N4,9,6:RETURN
```

Computer's game play strategy data.

```
6020 A$="1181188831611383168638683363
3
66641511484158548584353346435654656425
22474257547573262237326763767217 1"
6030 A$(LEN(A$)+N1)="12821787287822722
777":RETURN
```

Graphic bar chart for players' pieces.

```
7000 COLOR N0:PLOT N0,41:DRAWTO 79,41:
PLOT N0,44:DRAWTO 79,44
7010 IF (P2-SY)>N0 THEN COLOR OPPONENT
:PLOT N12,41:DRAWTO (P2-SY)+11,41
7020 IF SY>N0 THEN COLOR YOUR:PLOT N12
,44:DRAWTO SY+11,44
7030 RETURN
```

SWAT TABLE

For ATARI® FLIP-IT II

LINES	SWAT CODE	LENGTH	LINES	SWAT CODE	LENGTH
10 - 160	ZI	576	2020 - 3005	VG	509
170 - 250	IJ	536	3010 - 3052	QT	280
255 - 390	GX	341	3054 - 3100	PM	273
400 - 682	QT	381	3102 - 3150	ZY	322
700 - 812	IH	377	3152 - 4072	AY	476
820 - 910	ZA	442	4080 - 4162	GE	526
920 - 1040	DG	449	4170 - 4220	ZF	505
1045 - 1085	GM	387	4230 - 5040	HE	573
1090 - 1125	VQ	398	5050 - 5100	VM	534
1130 - 2000	WF	411	5110 - 7030	RL	492
2001 - 2014	NH	440			

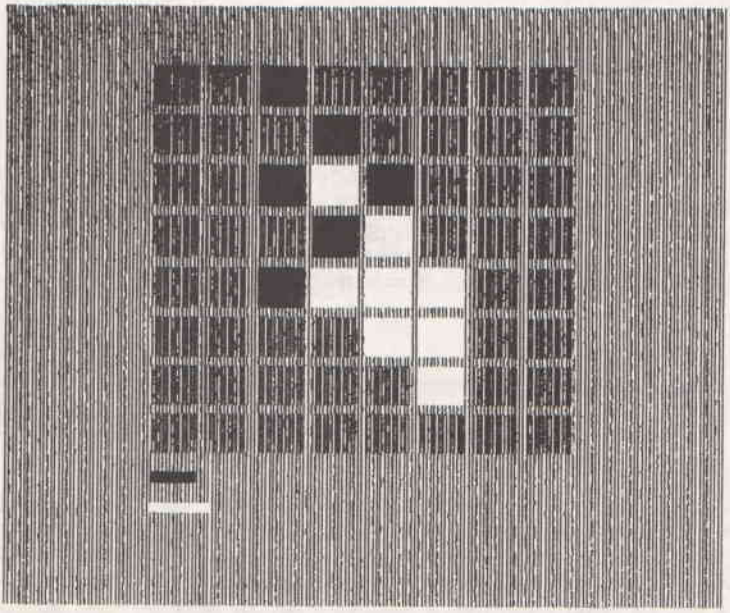


Figure 7. Another Atari® Flip-It screen.

General Information

These are the standard procedures for the programs published by **SoftSide** Publications, Inc. 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

At the conclusion of each **SoftSide** program listing, we include a **SWAT (Strategic Weapon Against Typos)** Table. **SWAT** for the Apple, Atari, and TRS-80 appeared in **SoftSide** Issue #30. **SWAT** for the IBM-PC appeared in **SoftSide** Issue #35. If you don't have **SWAT**, we'll send you a free reprint. Send a self-addressed, stamped envelope to:

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

Be sure to tell us what kind of computer you have.

Line Listings

IBM® PC USERS

The line listings in this booklet appear in a 40-column format. If you type LIST when your computer is displaying 40-column text (WIDTH 40), your screen should display exactly what appears in the printed listing. Be sure to use **SWAT** on your program, and get the free reprint if you don't have **SWAT**.

APPLE® USERS

The line listings in this booklet are in standard Applesoft™ format, and they appear exactly as they should on your screen when you type LIST. Things to watch out for when typing are:

- REM and DATA statements: Applesoft always LISTs an extra space after the keyword. Therefore, you should type one space less than what appears in the listing.

• Spaces between quotes: Applesoft is a bit eccentric about how it LISTs these. Just LIST the line after you type it, and compare it to the printed listing.

Also, be sure to use **SWAT** on your program, and get the free reprint if you don't have **SWAT.**

ATARI® USERS

The line listings in this booklet 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 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.

TRS-80® USERS

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. Be sure to use **SWAT** on your program, and get the free reprint if you don't have **SWAT.**

Copyright ©

This program is for your personal use only. Please resist the urge to give away copies of copyrighted material.

Notes

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:

NOW AVAILABLE ON DISK

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®		
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 _____

SUBSCRIBE!

If the high price of commercial software and the lack of clear information about your microcomputer has got you down, here's the solution you've been waiting for!

SoftSide Magazine

SoftSide is a favorite of computer users and hobbyists alike. **SoftSide** gives you the BASIC code listings of several programs — adventures, utilities, games, simulations, you name it — for your computer EVERY MONTH.

Send the coupon below to: **SoftSide Publications, Inc., 6 South Street, Milford, NH 03055.**

EITHER

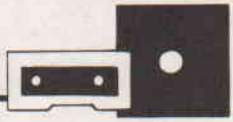
SUBSCRIBE TO SoftSide Magazine & *SoftSide* Selections

Please enroll me as a *SoftSide* subscriber. ☐ 6 issues for $18 ☐ 12 issues for $30

Please include the ☐ Apple® ☐ IBM®PC ☐ Atari® ☐ TRS-80® ***SoftSide* Selections Booklet.**

OR

SUBSCRIBE TO SoftSide & *SoftSide* Selections on Disk or Cassette



With *SoftSide* CV or DV you'll get the magazine plus each month's Apple®, IBM® PC, Atari®, or TRS-80® programs delivered on cassette or disk. All programs are tested and ready to run the day you receive them!

6 Issues	12 Issues
☐ *SoftSide* CV — $55	☐ *SoftSide* CV — $99
☐ *SoftSide* DV — $79	☐ *SoftSide* DV — $149

☐ 12 issues *SoftSide* DV — Four monthly payments of $38.50 (includes $5 billing charge).

Which computer? ☐ Apple® ☐ IBM® PC(**Disk Only**) ☐ Atari® ☐ TRS-80® Mod I ☐ TRS-80® Mod III

Name _____

Address _____

City/State _____ Zip _____

☐ **MasterCard** ☐ **VISA**

Name of Cardholder _____

MC# and Interbank#/VISA# _____

Exp. Date _____

Signature _____

Prices for USA orders only.

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.

FLIP-IT II

Flip-It II

You know your computer's smart — here's a chance to find out just how smart.

Reversi — the challenging board game of position and strategy — comes to your Apple®, Atari®, IBM® PC, and TRS-80® with **Flip-It II,** by Michael Prescott, with translations by the SoftSide programming staff.

Can you outflank your computer's pieces, or will its electronic intelligence see that opening first?

SOFTSIDE PUBLICATIONS, INC.
Milford, New Hampshire