

November 1986

Canada \$4.95 U.S. \$3.95

Just for your  
Tandy Color  
Computer 1, 2 and 3

The

# RAINBOW<sup>®</sup>

THE COLOR COMPUTER MONTHLY MAGAZINE

## *Ma Belle Amie*

### **The Beauty of Telecommunications**

Add an editor and graphics screens to CoBBS

Connect a second CoCo to your BBS setup

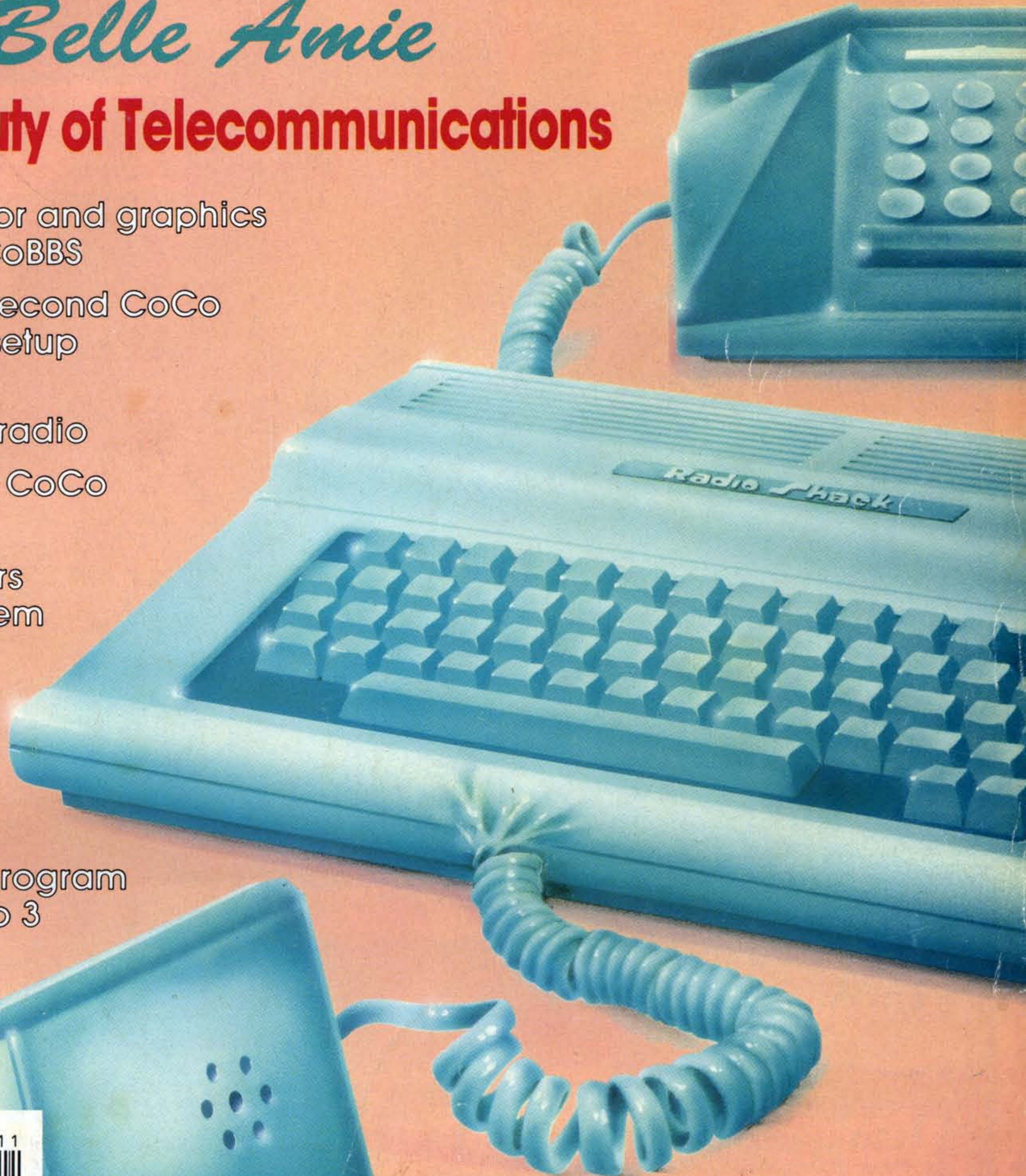
Ham it up with packet radio

Access your CoCo by remote

Play checkers over a modem

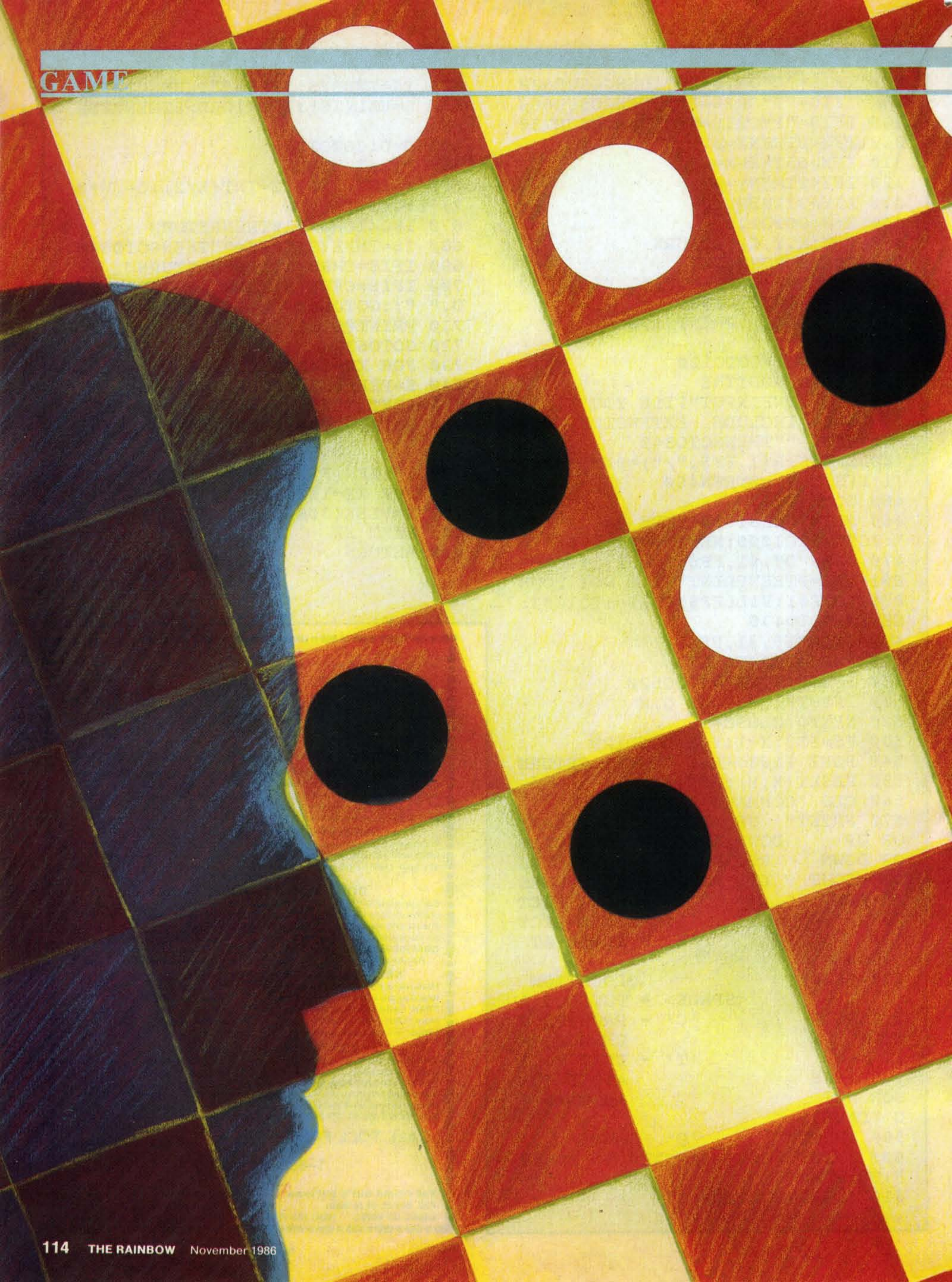
*Also,*

A terminal program for the CoCo 3



**And Doctor ASCII Joins the Rainbow**







*Checkers with a modem offers a new twist for an old favorite*

## Long Distance Draughts

By Greg Miller and Erik Gavriluk

**E**ach day more and more CoCo users are becoming interested in telecommunications and are purchasing modems to explore this exciting new world.

We are proud of our new program, *McCheckers*, which combines both modem programming and some of the graphics programming tricks we learned while writing *McPaint*. We are also very pleased to be able to share this program with a larger audience than was possible before, thanks to THE RAINBOW's support.

---

*Greg Miller, 18, is a college freshman majoring in electronic engineering. Erik Gavriluk, 15, is a high-school junior.*

### The Program

*McCheckers* is a machine language checkers game two people play over the modem. This means any two people having this program and a modem can play, whether they live across town, or across the country.

To make the file necessary to play *McCheckers*, you need to use two programs. The first, shown in Listing 1, is a BASIC program that draws the graphics checkerboard on which the game is played. Type in and run this program. After the display is generated, press any key to save the graphics screen. Be sure to save a copy of the BASIC program as well.

Listing 2 is a BASIC program to generate the machine language checkers



game. If you get the Checksum Error message, check the data lines, because it is likely that one or more of them contain an error. Also be sure to save a copy before you run the program; an error in typing could crash the computer.

After running Listing 2, save the completed program on cassette by typing CLOADM"CHECKBORD", &H800 and press ENTER. Then type CSAVEM"CHECKERS", &HE00, &H3300, &H2600 and press ENTER. For disk, type LOADM"CHECKBORD" and press ENTER. Then type SAVEM"CHECKERS", &HE00, &H3300, &H2600 and press ENTER.

### How to Play

Load the game and type EXEC. You will see a banner, along with the prompt "Originate or Answer?" The person using the answer mode on his modem should use Answer; the other person should use Originate. The person using Originate goes first.

Next, you are put into the type mode, where you can send commands to your modem (if it responds to commands like

a Hayes Smartmodem). If you have not already done so, you must now establish carrier between you and your opponent. Press BREAK to begin the game.

Both players move the white pieces on the bottom of the board. The program automatically displays the other player's pieces as black.

*McCheckers* is a complete implementation of checkers; the usual rules apply. Here's a brief overview:

- Pieces only move diagonally forward. A piece may be moved backward only if it is a king. A piece becomes a king when it reaches the last row of the opposing player (the top row on the screen).
- A piece must "jump" if at all possible. (This is an official rule of checkers, but is most often ignored in casual play.)
- The game ends when one player has captured all his opponent's pieces, or when a player has no possible move. If a player has no possible move, then the other player wins.

To move a piece, point the arrow to

the piece you want to move, and then to the destination square. If you make an illegal move, you are told so. You can only move a piece when the arrow appears on the screen. If the arrow does not appear on your screen, it means that the other player is in the process of moving. You must wait for the arrow to appear before you can move. When it is your turn (the arrow appears on the screen), you may send a short message to the other player by pressing CLEAR and then typing your message. Messages are displayed on the top line of the screen. If you receive a message, press the joystick button after reading the message; the other player will not be able to continue his turn until after you have done so.

At the end of a game each player is notified as to whether he won or not, and is again put into the type mode, where pressing BREAK begins a new game.

*(Questions about this program may be directed to the authors at 3101 Link Road #32, Lynchburg, VA 24503. Please enclose an SASE for a reply.)* □



### Listing 1: MCDRAW

```

1 ' BASIC PROGRAM TO DRAW
2 ' CHECKERBOARD FOR McCheckers
3 '
10 PMODE 4,1:PCLS1:SCREEN 1,1
20 DIM B(500),B2(500)
30 FOR Y=0 TO 30 STEP 6
40 LINE(0,Y)-(255,Y),PRESET
50 LINE(0,Y+1)-(255,Y+1),PRESET
60 NEXT Y
70 X1=58:Y1=45:X2=195:Y2=180
80 LINE(X1,Y1)-(X2,Y2),PRESET,B
90 LINE(X1+1,Y1+1)-(X2-1,Y2-1),PRESET,B
100 LINE(62,48)-(191,177),PRESET,BF
110 GET(0,40)-(13,53),B
120 FOR X=64 TO 190 STEP 32
130 FOR Y=50 TO 160 STEP 32
140 PUT(X,Y)-(X+13,Y+13),B,PSET
150 NEXT Y,X
160 FOR X=80 TO 176 STEP 32
170 FOR Y=66 TO 176 STEP 32
180 PUT(X,Y)-(X+13,Y+13),B,PSET
190 NEXT Y,X

```

```

200 FOR Y=32 TO 44
210 IF Y/2=INT(Y/2) THEN A=204 ELSE A=51
220 LC=&HE00+Y*32
230 FOR T=0 TO 31:POKE LC+T,A:NEXT T
240 NEXT Y
250 GET(0,33)-(255,44),B
260 PUT(0,181)-(255,192),B
270 GET(0,32)-(57,44),B,G
280 GET(196,32)-(255,44),B2,G
290 FOR Y=32 TO 180 STEP 12
300 PUT(0,Y)-(57,Y+12),B,PSET
310 PUT(196,Y)-(255,Y+12),B2,PSET
320 NEXT Y
330 FOR Y=4 TO 26
340 LC=&HE00+Y*32
350 FOR A=7 TO 24
360 READ B:POKE LC+A,B:NEXT A
370 NEXT Y
380 A$=INKEY$:IF A$="" THEN 380
390 CLS:PRINT"SAVING..."
440 A=PEEK(&HC000)
450 IF A=68 THEN SAVEM"CHECKBORD",&HE00,&H25FF,&HA027:END
460 CSAVEM"CHECKBOARD",&H600,&H1DFF,&HA027:END
470 DATA 255,252,15,255,255,192,24,31,255,255,255,255,255,255,255,255,255
480 DATA 255,249,136,31,255,31,3

```



```

, 31, 255, 255, 255, 255, 255, 255, 255,
255, 255, 255
490 DATA 255, 243, 131, 31, 254, 113,
134, 31, 255, 255, 255, 192, 255, 255, 2
55, 255, 255, 255
500 DATA 255, 247, 135, 31, 252, 192,
198, 31, 255, 255, 255, 216, 255, 255, 2
55, 255, 255, 255
510 DATA 255, 231, 143, 31, 249, 128,
204, 31, 255, 255, 255, 152, 255, 255, 2
55, 255, 255, 255
520 DATA 255, 237, 155, 24, 3, 0, 204,
63, 192, 15, 0, 176, 252, 0, 3, 255, 255,
255
530 DATA 255, 205, 155, 3, 195, 5, 140
, 1, 159, 132, 120, 48, 249, 248, 96, 6, 0
, 127
540 DATA 255, 217, 179, 14, 102, 12, 1
2, 240, 48, 193, 204, 48, 243, 12, 55, 19
2, 252, 63
550 DATA 255, 153, 179, 24, 54, 28, 25
, 152, 96, 99, 6, 48, 6, 6, 60, 99, 134, 63
560 DATA 255, 177, 227, 48, 54, 28, 27
, 12, 96, 102, 6, 96, 198, 6, 56, 6, 6, 63
570 DATA 255, 49, 230, 48, 102, 63, 21
9, 12, 192, 198, 12, 97, 140, 12, 48, 12,
12, 63
580 DATA 255, 97, 198, 96, 6, 63, 222,
12, 195, 140, 0, 103, 12, 56, 48, 12, 0, 6
3
590 DATA 255, 97, 198, 97, 6, 63, 222,

```

```

12, 222, 12, 32, 108, 13, 224, 96, 231, 1
92, 63
600 DATA 254, 97, 134, 97, 6, 63, 222,
13, 240, 12, 32, 120, 31, 0, 97, 224, 120
, 127
610 DATA 254, 193, 134, 99, 246, 31, 2
20, 12, 192, 12, 126, 240, 12, 0, 97, 224
, 12, 63
620 DATA 254, 192, 12, 99, 243, 28, 28
, 24, 192, 44, 126, 216, 76, 2, 99, 240, 6
, 63
630 DATA 252, 192, 12, 98, 3, 1, 156, 2
4, 192, 12, 64, 204, 12, 0, 195, 224, 6, 6
3
640 DATA 253, 132, 12, 96, 193, 195, 2
4, 24, 192, 204, 24, 198, 12, 12, 195, 23
6, 6, 63
650 DATA 253, 135, 236, 49, 128, 126,
24, 48, 99, 134, 48, 195, 6, 56, 195, 231
, 12, 63
660 DATA 253, 135, 224, 31, 0, 0, 24, 4
8, 62, 3, 224, 193, 131, 224, 199, 225, 2
48, 63
670 DATA 252, 15, 224, 0, 4, 0, 0, 128,
0, 0, 0, 0, 0, 0, 7, 224, 0, 63
680 DATA 252, 15, 224, 0, 15, 0, 0, 129
, 0, 0, 0, 0, 0, 0, 7, 240, 0, 127
690 DATA 252, 15, 255, 192, 31, 255, 1
92, 129, 128, 56, 2, 4, 8, 2, 7, 252, 0, 25
5

```

201	.....	221	216	.....	153
205	.....	48	219	.....	14
208	.....	133	223	.....	8
212	.....	86	END	.....	85

Listing 2: MCLDAD

```

1 ' BASIC loader for McCheckers
2 '
3 GOTO 10
4 GOTO 20
10 CLEAR 1000:PCLEAR 8:GOTO 4
20 CLS:AD=&H2600
30 FOR T=29 TO 1 STEP -1
40 PRINT T;
50 READ A$
60 Z$=LEFT$(A$,2):A$=MID$(A$,3)
70 V=VAL("&H"+Z$):CK=CK+V
80 POKE AD,V
90 AD=AD+1:IF A$<>" " THEN 60
100 NEXT T
110 PRINT:PRINT
120 IF CK<>285767 THEN PRINT"CHE
CKSUM ERROR" ELSE PRINT"DATA COR

```

```

RECT"
130 END
200 DATA BDA9287F09867FFF408E2C1
6BDB99CBDA1765F814F2706814126F4C
6FFF72C16F72C06BDA9288634B7FF038
E2C6ABDB99CBD2B6D8635B7FF03B62C1
6B72C06BD2FC4C6408E2B7D108E2BBDA
680A7A05A26F9CC0000FD2C14BD29C7A
6842603BD297FFC2C144C810826024F5
CFD2C14C10826E5BD
201 DATA 2B1A8E290FC60C3404EC81F
D2C144FBD295C6AE426F3C60CE7E4EC8
1FD2C148601BD295C6AE426F23261732
C067D2C061026008D8601C603FD2C00B
D2A7F10270605BD2AB3102705FE7D315
82AFB7F3158B6315B812126037E2E25C
C0707B0315DF0315EFD2C14BD29C7A68
46F84B72C0FBD297F
202 DATA CC0707B0315FF03160FD2C1
4BD29C7B62C15810726058603B72C0FB
62C0FA7844ABD295CFC31614D102BFF8
A3406CC0707A0E0E0E0FD2C14BD29C76
F84BD297F7D31631027FF6F732C067E2
69ECC0204FD2C00BD2A7F1027059FBD2
AB310270598BD274F7E269E8E2764BF2

```











