Thank you for buying Orion Software. We hope that your use of our product will be trouble free. To forward these ends, please give these instructions at least a cursory reading. Some pieces of Orion Software have unique features in their loading and use that you should be aware of.

****
* WARNING! *
****

Orion Software is recorded on <u>CHROME TAPE</u> because it will give you a more reliable load. However, it precludes your re-recording a program or putting another program on this tape with a standard portable cassette recorder (the type most of us use). If you attempt to re-record or record anything else and put it on your Orion tape, you will not succeed and you will undoubtedly damage the programs that are already there. If you do this your tape is not returnable.

We suggest that once you have loaded your Orion Software tape, and checked its functioning; that you make a backup copy on your own tape. Then put your original Orion tape away for use only should your backup copy(s) become unusable.

COPYRIGHT NOTICE

All software (either in machine readable or human readable form) and all instructions, listings, illustrations, descriptions, etc. included with it, sold by Orion Software Associates, Inc. is protected by copyright. All of these materials are sold under the express understanding that they may not be duplicated except for the purchaser's own personal use on personally owned system(s). To do otherwise is a violation of the law.

Educational and other institutions who wish to make copies for use on commonly owned systems please contact us.

RETURN POLICY

We don't use any cute protection gimmicks on Orion Software, so you can use your machine to make backup copies. However, you could also make a copy and then return the original tape (disk) requesting a refund or exchange for another program.

Unfortunately, not so. We will exchange proven defective tapes (disks) with <u>copies of the same program(s)</u> only if they meet these conditions:

1) The program doesn't load on your machine.
2) The program tape (disk) has not been tampered with (see paragraph headed WARNING!)
3) You pay postage both ways to return the tape (disk) to and from Orion Software Associates.
4) The program will not load on our machines either.

Our tapes are duplicated by professionals, using a digital process, under strict quality control. They are the best that can possibly be produced. Occasionally, tapes will have one or two bad characters. If this is the case (i.e. only one or two garbled characters) we do not consider this a bad loading tape. Listings are included with most tapes so you can correct this. On the other hand, if you are getting several garbled lines, this may be a bad tape. Such a tape should be returned to us. You must send postage for the return of your tape or a stamped, self-addressed, <u>padded</u> envelope. Be sure the postage is sufficient.

LOADING INSTRUCTIONS

BOARD GAMES 1

CUBIC

Cubic is written in BASIC and loads in the conventional manner. For those of you who may be new to the hobby, lets go through the loading procedures.

1) Be sure your system is correctly set up. Make sure the computer, monitor and cassette machine are interconnected corectly. See your owner's manual for directions.
2) Be sure your 'SHIFT LOCK' key is down! You'd be surprised how many people forget this.
3) Insert your Orion cassette into your recorder with the side you want up and rewound.
4) Cold start your system, or if it is already running, type 'NEW' (RETURN).
5) Type 'LOAD' but do not hit (RETURN).
6) Push play on your recorder.
7) Now hit (RETURN).
8) When loading stops, stop the cassette, and push the SPACE BAR then (RETURN).
9) Type 'RUN' (RETURN).

MINI-GOMOKU

Mini-Gomoku is written in machine language but loads via BASIC. It does not POKE in the machine language routine, but uses a special loading technique. To load Mini-Gomoku follow steps 1 to 7 above. As the program loads you will see a few lines of BASIC, then an O.K., then one more line of BASIC. Finally you will see a number at the lower left of the screen being incremented rapidly. This is the program being loaded. If you do not see this, start over, this time readjusting your tape machine's volume and tone controls until that number increments rapidly and smoothly. When the program has finished loading, it will auto start and the playing board will scroll onto the screen.

NOTES TO TV USERS

The graphics displays of these games have been developed using true video monitors. We have made every effort to use only that portion of the screen that can also be seen by those people using a converted TV (or a TV with an RF converter). However, we cannot be sure that the playing boards will show in totality on all converted TV's, as each one seems to show more or less of the screen than others. We apologise for any inconveince.

COLOR AND SOUND

Since these games are games of intellect, color and sound were deemed unnessary even distracting.

PLAYING INSTRUCTIONS

BOARD GAMES 1

CUBIC

Cubic is a game of three dimensional tic-tac-toe. It is played on a 4x4x4 board because, if you have ever tried to play on a 3x3x3 board, you know that it makes the game trivial.

What is meant by a 4x4x4 board? Each level has 4 rows of 4 columns and there are 4 levels. Rows go horizontally across the level with row one at the top and row 4 at the bottom (see fig. 1). Columns run up and down on a level with column 1 at the left and column 4 at the right. The levels are numbered 1 to 4 starting with the uppermost.

While the program does give a graphics display of the board it is intended primarily to check piece positions. To give yourself the best idea of the progress of the game we recommend that you build or buy a playing board. The Qubic^{tm} board by Parker Bros. is one such board that you can buy.

The computer will open the game by asking if it should move first. If you want it to move first answer affirmatively. For those of you who are not used to OSI machines, you must give some answer to every promt for one. If you answer with a RETURN only it will terminate the program. So if you want to move first, answer negatively. If the computer moves first, it will ask for a random number seed. This number must be a decimal. It will then output its first move; level, row and column.

At this point, or if you elected to move first, the computer will ask for your move. Enter the level of your intended move first and then a RETURN. It will then ask you for the row; again, enter the row followed by a RETURN. It will then ask for the column; follow the same procedure.

If you have entered any one of these numbers wrong and HAVE NOT YET HIT RETURN FOR THAT NUMBER, simply enter an illegal number (e.g. any number greater than 4), and you will be asked again. After you have hit RETURN for a level, row or column, you cannot practically change that entry.

If the square you have requested is taken, you will be told and the entry process will be started over. Herein lies the only practical way to change a move that has been partially entered. If you have entered a level, or a level and row, try to manipulate your remaining entries so that the square requested is already taken. You will then be started over.

Once you have sucessfully entered a move the computer will draw the board for you to view and ponder while it considers its next move. The computer will return with its move in about 90 seconds. When it does it will clear the screen, display its move and query you for yours. This procedure is better than just placing a piece on the board and leaving you to figure out where the new piece is.

Oh, by the way, you are the white pieces and the computer is the black regardless of who moves first.

The game alternates in this way until one side wins or each side has made twenty moves, whichever comes first.

LOADING INSTRUCTIONS

BOARD GAMES 1

CUBIC

Cubic is written in BASIC and loads in the conventional manner. For those of you who may be new to the hobby, lets go through the loading procedures.

1) Be sure your system is correctly set up. Make sure the computer, monitor and cassette machine are interconnected corectly. See your owner's manual for directions.
2) Be sure your 'SHIFT LOCK' key is down! You'd be surprised how many people forget this.
3) Insert your Orion cassette into your recorder with the side you want up and rewound.
4) Cold start your system, or if it is already running, type 'NEW' (RETURN).
5) Type 'LOAD' but do not hit (RETURN).
6) Push play on your recorder.
7) Now hit (RETURN).
8) When loading stops, stop the cassette, and push the SPACE BAR then (RETURN).
9) Type 'RUN' (RETURN).

MINI-GOMOKU

Mini-Gomoku is written in machine language but loads via BASIC. It does not POKE in the machine language routine, but uses a special loading technique. To load Mini-Gomoku follow steps 1 to 7 above. As the program loads you will see ~~a few lines of BASIC, then an O.K., then one more line of BASIC. Finally~~ you will see a number at the lower left of the screen being incremented rapidly. This is the program being loaded. If you do not see this, start over, this time readjusting your tape machine's volume and tone controls until that number increments rapidly and smoothly. When the program has finished loading, it will auto start and the playing board will scroll onto the screen.

NOTES TO TV USERS

The graphics displays of these games have been developed using true video monitors. We have made every effort to use only that portion of the screen that can also be seen by those people using a converted TV (or a TV with an RF converter). However, we cannot be sure that the playing boards will show in totality on all converted TV's, as each one seems to show more or less of the screen than others. We apologise for any inconveince.

COLOR AND SOUND

Since these games are games of intellect, color and sound were deemed unnessary even distracting.

	COLUMN 1	COLUMN 2	COLUMN 3	COLUMN 4
ROW 1				
ROW 2		e.g. Row 2, Col. 2		
ROW 3				
ROW 4				

SPECIAL TECHNIQUES

CUSTOMIZING

The playing pieces displayed are stored as WT and BT for the white and black pieces, respectively. By changing their assignment statement (line 270) you can make them any graphics character.

Twenty was chosen semi-arbitrarily as the maximum number of moves in a game. MC holds the number of moves made. By changing the test on MC in line 1400 you can increase or decrease this number.

SPECIAL BOARD POSITIONS

No direct provision has been made for this in the program, but it can be done. The computer's image of the board is the matrix BD(4,4,4). The indices of this matrix are the level, row and column respectively.

Run the program, and break out of it when the computer asks who should move first by using just a RETURN in answer to this question. Then, using a series of immediate mode commands, set the squares you want to occupy in the BD matrix to 1 and the computer's squares to 5. Set MC to the number of moves that you want to have been used. (e.g. BD(1,2,3)=1...BD(4,3,2)=5...MC=6)

Now reenter the program by a GOTO 1107. DO NOT use RUN 1107, as this will reset the values in BD to zero. This enters via the graphics routine, which will draw the board while the computer generates its next move.

MINI-GOMOKU

This game is 'mini' because it plays on only a 10x10 board rather than the standard 16x16. This was dictated by considerations of some OSI systems', display capability. The graphics display of the board is totally sufficient to play the game; however, many confirmed Gomoku players like to use a real wooden board and 'stones.'

Gomoku is an ancient oriental game. The idea is to place five of your 'stones' (pieces) in a row and prevent your opponent from doing so. Volumes and volumes have been written on gomoku strategy so, for the uninitiated, they are the best place to learn the game.

As said in the loading instructions, the game auto starts on loading. You will be asked to move first, and you will, therefore, be the black stones. The game has no provision for the computer making the first move, nor is there an easy way to set up special board positions.

Enter your moves by entering a row, which have alphabetic designations (A-J); and then a column, which have numeric designations (0-9). Simply push the letter and number keys for your chosen square, no RETURN is necessary. Be careful, however, as there is no way to take back any key pushed.

After the second keystróke, your piece is placed; the computer figures its move (in less than one second); places its pieee and displays the alphanumeric move it has made on the bottom of the screen. Look quickly as this display lasts only about a second.

The game continues until one side wins, or a draw occurs. A draw is recognized by the computer when it is no longer possible for either side to win.

At the conclusion of the game, the computer will ask if you want to play again. Pushing 'Y' will gat you another game. If you should accidentally hit BREAK during a game push 'W' (warm start) then either type 'RUN' and RETURN or use X=USR(X) in the immediate mode. This will not save the game in progress, but it will save you from having to reload the program

MAKING COPIES OF 'MINI-GOMOKU'

There are two copies of 'Mini-Gomoku' on the tape you have received. But you may, and probably should, want to make baskup copies. We are not, at this time, releasing the program which puts machine language programs into the format that you received 'Mini-Gomoku' in.

Probably the best and easiest way to make machine language tapes, without OSI's extended monitor, was outlined by Danny Schwartz (Mini-Gomoku's author) in the April/May issue of 'Compute II.' Since we cannot print any part of that article here you willhave to get a copy of that magazine to use that method. You will need to know that Mini-Gomoku starts at 2276 and ends at 3747 decimal. You can reach 'Compute II' at P.O. Box 5119, Greensboro, N.C. 27403.

```
LIST

 10 IFPEEK(57088)<128THENPOKE56832,0
 110 FORZ=546TO570
 120 READM:POKEZ,M
 130 NEXTZ:POKE11,34:POKE12,2
 140 DATA162,216,169,208,133,255,169,0,133,254
 150 DATA168,169,32,145,254,200,208,251,230,255
 160 DATA228,255,208,245,96
 170 X=USR(X)
 180 PRINT"      CUBIC V 2.0":PRINT:PRINT:PRINT:PRINT:PRINT
 190 PRINT"       (C) 1980":PRINT
 195 PRINT"    ORION SOFTWARE":PRINT"    ASSOCIATES, INC."
 197 PRINT:PRINT:PRINT:PRINT:PRINT
 200 FORTT=1TO1E4:NEXT
 250 X=USR(X)
 260 DIMBD(4,4,4),DM(4,4,4),IM(4,3),TA(3)
 270 BA=53379:OF=32:WT=232:BT=226:BB=BA
 280 IFPEEK(57088)<128THENOF=64
 951 PRINT"THE GAME IS LIMITED TO":PRINT"TWENTY MOVES"
 952 PRINT:PRINT:PRINT
 955 PRINT"READY TO PLAY???"
 960 PRINT:PRINT:PRINT:PRINT:PRINT
 965 PRINT"SHALL I MOVE FIRST?"
 970 PRINT:PRINT:PRINT:PRINT:PRINT
 975 PRINT"(YES/NO)"
 980 INPUT A$
 985 IF A$="YES"OR A$="Y" THEN GOSUB 5000
 995 PRINT:PRINT:PRINT
 1000 PRINT"WHAT IS YOUR MOVE?"
 1005 PRINT"LEVEL":INPUT L
 1007 IF L>4 THEN 1005
 1010 PRINT"ROW":INPUT R
 1012 IF R>4 THEN 1010
 1015 PRINT"COLUMN":INPUTC
 1017 IF C>4 THEN 1015
 1019 IFBD(L,R,C)=0THEN1100
 1020 PRINT"THAT SQUARE IS TAKEN":GOTO1000
 1100 MC=MC+1
 1105 BD(L,R,C)=1
 1107 X=USR(X)
 1110 GOSUB8000
 1200 GOSUB 2000
 1300 GOSUB 4000
 1400 IF MC<20 THEN 1000
 1500 GOTO 6000
 2000 CO=1
 2010 FORL=1TO4
 2015 FOR R=1 TO 4
 2020 FOR C=1 TO 4
 2025 IF BD(L,R,C)=0 THEN GOSUB 2900
 2030 H=BD(L,R,C)+H
 2035 NEXTC
 2038 IFH=0THENGOSUB3700
 2040 IF H>0 THEN GOSUB3000
 2045 H=0:CO=1
 2050 NEXT R
 2060 H=0:CO=1
 2070 FOR C=1 TO4
 2075 FOR R=1 TO 4
 2080 IF BD(L,R,C)=0THEN GOSUB 2900
 2085 H=BD(L,R,C)+H
 2090 NEXT R
 2093 IF H=0THENGOSUB3700
 2095 IF H>0 THEN GOSUB 3000
```

```
2100 H=0:CO=1
2105 NEXTC
2110 H=0:CO=1
2115 FORR=1TO4
2120 C=R
2125 IFBD(L,R,C)=0THENGOSUB2900
2130 H=BD(L,R,C)+H
2140 NEXTR
2145 IFH=0THENGOSUB3700
2150 IFH>0THENGOSUB3000
2155 H=0:CO=1
2160 FORR=1TO4
2165 C=5-R
2170 IFBD(L,R,C)=0THENGOSUB2900
2175 H=BD(L,R,C)+H
2180 NEXTR
2185 IFH=0THENGOSUB3700
2190 IFH>0THENGOSUB3000
2195 H=0:CO=1
2197 NEXTL
2198 H=0:CO=1
2200 FORR=1TO4
2210 FORC=1TO4
2220 FORL=1TO4
2230 IFBD(L,R,C)=0THENGOSUB2900
2240 H=BD(L,R,C)+H
2250 NEXTL
2260 IFH=0THENGOSUB3700
2270 IFH>0THENGOSUB3000
2280 H=0:CO=1
2290 NEXTC
2295 NEXTR
2297 H=0:CO=1
2300 FORR=1TO4
2305 FORC=1TO4
2307 L=C
2310 IFBD(L,R,C)=0THENGOSUB2900
2315 H=BD(L,R,C)+H
2320 NEXTC
2325 IF H=0THENGOSUB3700
2330 IFH>0THENGOSUB3000
2335 H=0:CO=1
2340 NEXTR
2345 H=0:CO=1
2350 FORR=1TO4
2355 FORL=1TO4
2360 C=5-L
2365 IFBD(L,R,C)=0THENGOSUB2900
2370 H=BD(L,R,C)+H
2375 NEXTL
2380 IFH=0THENGOSUB3700
2385 IFH>0THENGOSUB3000
2390 H=0:CO=1
2395 NEXTR
2397 H=0:CO=1
2400 FOR C=1TO4
2405 FOR R=1TO4
2410 L=R
2415 IFBD(L,R,C)=0THENGOSUB2900
2420 H=BD(L,R,C)+H
2425 NEXTR
2430 IF H=0THENGOSUB3700
2435 IFH>0THENGOSUB3000
2440 H=0:CO=1
2445 NEXTC
2450 H=0:CO=1
```

```
2455 FORC=1TO4
2460 FORL=1TO4
2465 R=5-L
2470 IFBD(L,R,C)=0THENGOSUB2900
2475 H=BD(L,R,C)+H
2480 NEXTL
2485 IFH=0THENGOSUB3700
2490 IFH>0THENGOSUB3000
2492 H=0:CO=1
2495 NEXTC
2497 H=0:CO=1
2500 FORL=1TO4
2505 R=L:C=L
2510 IFBD(L,R,C)=0THENGOSUB2900
2515 H=BD(L,R,C)+H
2520 NEXTL
2525 IF H=0THENGOSUB3700
2530 IFH>0THENGOSUB3000
2535 H=0:CO=1
2550 FORL=1TO4
2555 R=L:C=5-L
2560 IFBD(L,R,C)=0THENGOSUB2900
2565 H=BD(L,R,C)+H
2570 NEXTL
2575 IFH=0THENGOSUB3700
2580 IFH>0THENGOSUB3000
2585 H=0:CO=1
2600 FORL=1TO4
2605 R=5-L:C=R
2610 IFBD(L,R,C)=0THENGOSUB2900
2615 H=BD(L,R,C)+H
2620 NEXTL
2625 IFH=0THENGOSUB3700
2630 IFH>0THENGOSUB3000
2635 H=0:CO=1
2650 FORL=1TO4
2655 C=L:R=5-L
2660 IFBD(L,R,C)=0THENGOSUB2900
2665 H=BD(L,R,C)+H
2670 NEXTL
2675 IFH=0THENGOSUB3700
2680 IFH>0THENGOSUB3000
2685 H=0:CO=1
2900 IM(CO,1)=L
2910 IM (CO,2)=R
2920 IM(CO,3)=C
2925 CO=CO+1
2930 RETURN
3000 IF H=15 GOTO 3100
3010 IF H=3 GOTO 3200
3020 IF H=2 GOTO 3300
3030 IF H=10 GOTO 3400
3035 IF H=4 GOTO 7500
3040 IF H=1ORH=5THENGOTO3700
3050 GOTO3700
3100 FOR F=1 TO4
3105 IF IM(F,1)=0 THEN 3700
3115 DM(IM(F,1),IM(F,2),IM(F,3))=100
3120 NEXT F
3130 GOTO 3700
3200 FOR F=1 TO4
3205 IF IM(F,1)=0 THEN3700
3215 DM(IM(F,1),IM(F,2),IM(F,3))=90
3220 NEXTF
3230 GOTO 3700
```

```
3300 FOR F=1 TO 4
3305 IF IM (F,1)=0 THEN 3700
3315 DM(IM(F,1),IM(F,2),IM(F,3))=DM(IM(F,1),IM(F,2),IM(F,3))+10
3320 NEXTF
3330 GOTO3700
3400 FOR F=1 TO4
3405 IF IM(F,1)=0THEN3700
3415 DM(IM(F,1),IM(F,2),IM(F,3))=DM(IM(F,1),IM(F,2),IM(F,3))+5
3420 NEXTF
3430 GOTO3700
3700 FOR F=1 TO4
3710 FOR S=1TO 3
3720 IM(F,S)=0
3730 NEXTS:NEXTF
3740 RETURN
4000 REM COMPUTER MOVE ROUTINE
4005 T=DM(1,1,1)
4010 TA(1)=1:TA(2)=1:TA(3)=1
4015 FORL=1TO4
4020 FORR=1TO4
4025 FORC=1TO4
4030 IFDM(L,R,C)>TTHEN4100
4035 GOTO4200
4100 T=DM(L,R,C)
4105 TA(1)=L
4110 TA(2)=R
4115 TA(3)=C
4200 NEXTC
4205 NEXTR
4210 NEXTL
4300 IFT=0THENGOSUB5057:GOTO4500
4400 L=TA(1)
4405 R=TA(2)
4410 C=TA(3)
4415 BD(L,R,C)=5
4420 GOSUB5500
4500 FORL=1TO4:FORR=1TO4:FORC=1TO4
4505 DM(L,R,C)=0
4510 NEXTC:NEXTR:NEXTL
4520 RETURN
5000 X=USR(X)
5005 PRINT"THERE ARE SO MANY "
5010 PRINT"CHOICES, I CAN'T DECIDE"
5015 PRINT"GIVE ME A HAND BY "
5020 PRINT"CHOOSING YOUR FAVORITE"
5025 PRINT"NUMBER--(0-1)"
5030 PRINT:PRINT:PRINT:PRINT:PRINT
5040 INPUT SE
5041 IF SE<1 THEN 5045
5042 PRINT"THE NUMBER MUST BE "
5043 PRINT"BETWEEN ZERO AND ONE"
5044 PRINT"THAT MEANS A DECIMAL!!!":GOTO 5040
5045 PRINT:PRINT:PRINT:PRINT:PRINT
5050 PRINT"THANKS--I NEEDED THAT!!"
5055 PRINT:PRINT:PRINT:PRINT
5057 FORQ=1TO100:SE=RND(SE):NEXT
5060 L=INT(10*RND(SE))
5065 IF L>4 OR L<1 THEN 5060
5070 R=INT(10*RND(SE))
5075 IF R>4 OR R<1 THEN 5070
5080 C=INT(10*RND(SE))
5085 IF C>4 OR C<1 THEN 5080
5087 IF BD(L,R,C)<>0 THEN 5060
5090 BD(L,R,C)=5
5095 GOSUB 5500
5100 RETURN
```

```
5500 REM MOVE OUTPUT ROUTINE
5503 X=USR(X)
5505 PRINT:PRINT:PRINT:PRINT:PRINT
5510 PRINT"MY MOVE IS---"
5515 PRINT"LEVEL  ";L
5520 PRINT"ROW    ";R
5525 PRINT"COLUMN ";C
5530 PRINT:PRINT:PRINT:PRINT:PRINT
5535 IF T>=100 THEN GOSUB 7000
5540 RETURN
6000 PRINT"WE HAVE USED ALL OUR "
6005 PRINT"MOVES. THE GAME IS"
6010 PRINT"A DRAW."
6015 GOTO8600
7000 PRINT"I BELIEVE I HAVE WON!!"
7003 PRINT:PRINT:PRINT
7005 PRINT"THANK YOU FOR A REWARD-":PRINT"ING GAME!!!"
7007 PRINT:PRINT:PRINT:PRINT:PRINT
7010 GOTO8600
7500 PRINT:PRINT:PRINT
7505 PRINT"I BELIEVE YOU WIN!!!"
7510 PRINT"THANK YOU FOR A REWARD-"
7515 PRINT"ING GAME."
7520 PRINT:PRINT:PRINT
7525 GOTO8600
8000 PRINT"Level 1         Level 2":PRINT:PRINT
8002 PRINT"Level 3         Level 4":PRINT:PRINT
8003 PRINT:PRINT:PRINT:PRINT:PRINT
8004 FORSF=1TO2
8005 FORSE=1TO2
8010 FORSC=1TO4
8015 FORSD=0TO9STEP3
8020 POKEBA+SD,210:POKEBA+SD+2,207:POKEBA+SD+1,135
8025 POKEBA+SD+OF,136:POKEBA+SD+OF+2,143
8030 POKEBA+SD+2*OF+2,208:POKEBA+SD+2*OF+1,128
8035 POKEBA+SD+2*OF,209
8040 NEXTSD
8050 BA=BA+3*OF:NEXTSC
8055 BA=BA+4*OF:NEXTSE
8060 BA=BB+14:NEXTSF
8070 BA=BB
8100 FORL=1TO4
8105 FORR=1TO4
8110 FORC=1TO4
8115 IFBD(L,R,C)<>0THEN8125
8120 GOTO8150
8125 ONLGOSUB8200,8300,8400,8500
8130 IFBD(L,R,C)=5THENPOKESF,BT
8135 IFBD(L,R,C)=1THENPOKESF,WT
8150 NEXTC:NEXTR:NEXTL
8155 RETURN
8200 SF=BB+(3*R-2)*OF+C*3-2
8205 RETURN
8300 SF=BB+(3*R-2)*OF+C*3+12:RETURN
8400 SF=BB+(3*R-2)*OF+C*3-2+16*OF
8405 RETURN
8500 SF=BB+(3*R-2)*OF+C*3+12+16*OF:RETURN
8600 INPUT"WANT TO PLAY AGAIN";A$
8620 IFA$="Y"ORA$="YES"THENCLEAR:GOTO250
8630 IFPEEK(57088)<128THENPOKE56832,1
8640 X=USR(X):PRINT"BYE-BYE FOR NOW!"
OK
```

CUBIC ERRATA

If you have a C1 you need read no further, this only applies to C2, C4, and C8 users. We're really embaressed about this one. The very last thing added to the Cubic program was the part of the graphics routine that scrolls up the ledgends- Level 1, level 2, etc. This was done on a C1 just like most of our software development because of the ease of transferring from disk to tape on a C1. The tape was then put aside to be checked on our C4 before mastering the Board Games ! tape for duplication. Well, Murphy's Law struck. As time grew short and the first ads were about to appear; our C4's video board gave up the ghost. While it was out for repairs, we decided to go ahead and send the tapes out to be mastered. After all, we thought, we had checked everything many times on the C4.

Matters were compounded when the trial tapes came back for approval, the C4 was out again; this time for a non-functioning cassette port. So the tapes were approved and the first production run was made.

Needless to say, when they came back we discovered one major error. The ledgends I mentioned before did not scroll up enough on the C4 screen. They were also slightly off center.

The fix is a simple one and requires that you retype only two lines (only one if you can live with slightly off center titles. The two lines are shown below as they should be retyped. The changes from the existing lines ammounts to adding three spaces before the first title on each line, and adding three more PRINTS to line 8002. If you do not elect to add the three spaces to fix the centering, just retype line 8002 to add the PRINT's (there must be six total) and you are done.

If you do add the spaces line 8002 becomes rather long. You then have to add the PRINTs to line 8002 by using the shorthand form of ?:?:?:?:?:?. The machine will expand this to the necessary six PRINT statements. All of the last word **PRINT** might not show when the line is listed, but it will work anyway.

When you go to save this long line on tape, however, you may have trouble unless your terminal width had been set to 80 on start up. If this is the case, POKE 15,80 to set the terminal width to 80.

We are very sorry about this and it is not indicative of the quality of software we intend to produce.

```
PATCH FOR C-2, C-4, AND C-8 SYSTEMS. CUBIC GAME


 8000 PRINT"    LEVEL 1         LEVEL 2":PRINT:PRINT
 8002 PRINT"    LEVEL 3         LEVEL 4":PRINT:PRINT:PRINT:PRINT:PRINT:PRINT

OK
```