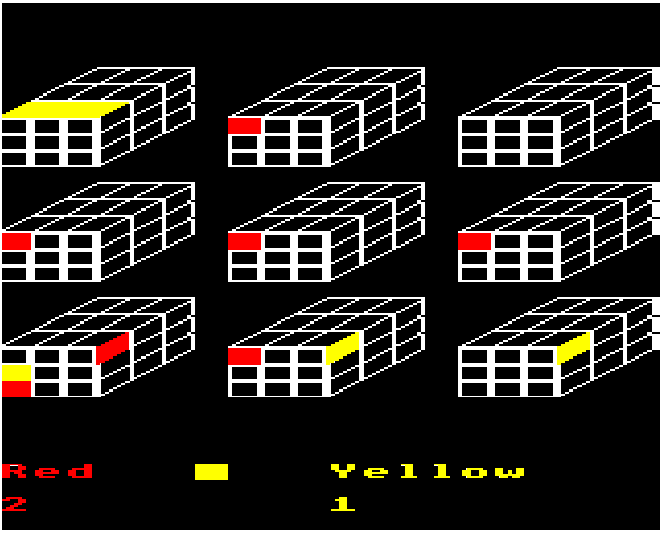
3D BRAINSTORM



If you feel like some brainstorming, how about this three-dimensional game for two players. It is like three-dimensional noughts and crosses, except it has the advantage that all faces can be seen at a glance.

Nine cubes are displayed on the screen and each face of each cube is divided into nine squares. Your aim is to colour in a row or column of squares in which you have coloured in the same square. For example, in the screen show above, the red player has coloured in a row of squares on the top left hand cube. Also the yellow player has coloured in all the top left-hand squares on the faces on the front of the central cube, so he has both a row and column of squares coloured in.

How to play

Players should take turns to colour in a square. The first player will colour in red, and the second in yellow. Squares are coloured in by moving the cursor (flashing underline symbol) to point to the correct square using the arrow keys.

If the square is on the front or top face of a cube, it is pointed to by putting the cursor under the square. If it is on the side of a cube it is pointed to by putting the cursor in the centre of the square. When the correct square has been selected, the space bar must be pressed and the appropriate square will be coloured in.

The score is shown at the bottom of the screen. The colour of the square between the titles indicates whose turn it is to play.

If you complete a row or column the computer makes an arcade-like warbling sound. If you try to colour in a square that does not exist a low constant note sounds.

Programming notes

The logic for a game like this is more complicated than you would expect because of the number of possible moves. The program is made more comprehensible and the program length reduced by using functions. These functions, however, do not deal with numbers but with logical operators. For example, in line 160 FNX will be true if XX is less than 0 or greater than 20. This is a check that regularly has to be done to see if the X-coordinate of the object would fit on the screen.

You could make this into a one player game by allowing the computer to make random guesses. You would do this by putting IF T=-1 THEN in front of line 520. You may also add the extra line IF T=1 THEN PS=RND(21)-1 : VP=RND(24)

```
    10 REM 3D BRAINSTORM
    20 REM COPYRIGHT (C) G.LUDINSKI 1983
    30 MODE 5
    40 DIM L%(20,24),M%(20,24),SC(2),R(6,
2)
    50 REM
    60 REM LOGICAL FUNCTIONS
    70 REM
    80 DEFFNL1(I)=(PS=0+I OR PS=1+I OR PS
=2+I OR PS=7+I OR PS=8+I OR PS=9+I OR PS
=14+I OR PS=15+I OR PS=16+I)
    90 DEFFNL2(I)=(VP=7+I OR VP=8+I OR VP
=9+I OR VP=14+I OR VP=15+I OR VP=16+I OR
 VP=21+I OR VP=22+I OR VP=23+I)
   100 DEFFNL7(P,I)=(P=I OR P=I+7 OR P=I+
14)
   110 DEFFNLP(I,J)=(IF PS=I AND VP=J)
   120 DEFFNF=(FNL1(0) AND FNL2(0))
   130 DEFFNT=((FNL1(0) AND (VP=6 OR VP=1
3 OR VP=20)) OR (FNL1(1) AND (VP=5 OR VP
=12 OR VP=19)) OR (FNL1(2) AND (VP=4 OR
VP=11 OR VP=18)))
   140 DEFFNE=(FNL7(PS,6) ORFNL7(VP,10) O
R(FNL7(PS,0) AND(FNL7(VP,4) ORFNL7(VP,5)
)) OR(FNL7(PS,1) ANDFNL7(VP,4)) OR (PS<0
 OR PS>20 OR VP<4 OR VP>24))
   150 DEFFNE2=(((FNL7(PS,3) OR FNL7(PS,4
) OR FNL7(PS,5)) AND FNL7(VP,9)) OR ((FN
L7(PS,4) OR FNL7(PS,5)) AND FNL7(VP,9))
OR ((FNL7(PS,4) OR FNL7(PS,5)) AND FNL7(
VP,8)) OR (FNL7(PS,5) ANDFNL7(VP,7)))
   160 DEFFNX(XX)=(XX<0 OR XX>20)
   170 DEFFNY(YY)=(YY<4 OR YY>24)
   180 DATA1,2,1,-1,-1,-2,7,14,7,-7,-7,-1
4
   190 FORI=1TO6:READ R(I,1):READ R(I,2):
NEXT
   200 REM
   210 REM SHAPE DEFINITIONS
   220 REM
   230 VDU23,224,&01,&02,&04,&08,&10,&20,
&40,&FF:REM /_
   240 VDU23,225,&01,&02,&04,&08,&10,&20,
&40,&80:REM /
   250 VDU23,226,&FF,0,0,0,0,0,0,0:REM -
   260 VDU23,227,&81,&82,&84,&88,&90,&A0,
&C0,&80:REM 1/
   270 VDU23,228,&80,&80,&80,&80,&80,&80,
&80,&80
   280 VDU23,229,&FF,&81,&81,&81,&81,&81,
&81,&FF:REM LI
   290 VDU23,230,&01,&03,&05,&09,&11,&21,
&41,&81:REM /1
   300 VDU23,231,&82,&83,&85,&89,&91,&A1,
&C1,&81:REM 1/1
   310 VDU23,232,&FF,&FE,&FC,&F8,&F0,&E0,
&C0,&80:REM TOP DIAG
   320 VDU23,233,&01,&03,&07,&0F,&1F,&3F,
&7F,&FF:REM LOWER DIAG
   330 VDU23,234,&FF,&FF,&FF,&FF,&FF,&FF,
&FF,&FF:REM FILLED IN
```

```
  340 L$=CHR$(224):D$=CHR$(225):T$=CHR$(
226):V$=CHR$(227):U$=CHR$(228):S$=CHR$(2
29)
  350 DU$=CHR$(230):VU$=CHR$(231):D1$=CH
R$(232):D2$=CHR$(233):F$=CHR$(234)
  360 FORJ=4TO24:FORI=0TO20:L%(I,J)=0:M%
(I,J)=0:NEXT:NEXT
  380 REM
  390 REM DRAW CUBES
  400 REM
  410 COLOUR BL+B:COLOUR WH:CLS
  420 FORY0=3 TO 17 STEP 7
  430   FOR X0=0 TO 14 STEP 7
  440     PROC_CUBE(X0,Y0)
  450   NEXTX0
  460 NEXTY0
  470 VDU30
  480 T=1:SC(0)=0:SC(2)=0
  485 COLOUR1:PRINTTAB(0,28)"Red   ™   "
;:COLOUR2:PRINT"Yellow":COLOUR1:PRINTTAB
(0,30)"0";:COLOUR2:PRINT"         0"
  490 REM
  500 REM MAIN CYCLE
  510 REM
  520 I$=GET$:PS=POS:VP=VPOS
  530 IF FNE OR FNE2 THEN SOUND1,-15,50,
10:GOTO520
  540 IF ABS(M%(PS,VP))=1 THEN SOUND1,-1
5,50,10:GOTO520
  550 T=-T:IF T=-1 THEN COLOUR1 ELSE COL
OUR2
  560 IF FNF THEN PROC_PUT(PS,VP,F$):GOT
O590
  570 IF FNT THEN PROC_PUT(PS,VP,D2$):PR
OC_PUT(PS+1,VP,D1$):GOTO590
  580 PROC_PUT(PS,VP,D2$):PROC_PUT(PS,VP
+1,D1$)
  590 M%(PS,VP)=T:VDU30
  600 PROC_TEST:COLOUR1+(-T+1)/2:PRINTTA
B(6,28);F$
  610 COLOUR1:PRINTTAB(0,28)"Red":COLOUR
2:PRINTTAB(10,28)"Yellow"
  620 COLOUR1:PRINTTAB(0,30);SC(0);:COLO
UR2:PRINTTAB(10,30);SC(2)
  630 GOTO520
  640 REM
  650 DEFPROC_CUBE(X0,Y0)
  660 PRINTTAB(X0,Y0)"   ___"
  670 PRINTTAB(X0,Y0+1)"  ";L$;L$;L$;DU$
  680 PRINTTAB(X0,Y0+2)" ";L$;L$;L$;D$;V
U$
  690 PRINTTAB(X0,Y0+3)L$;L$;L$;D$;V$;VU
$
  700 PRINTTAB(X0,Y0+4)S$;S$;S$;V$;V$;V$
  710 PRINTTAB(X0,Y0+5)S$;S$;S$;V$;V$
  720 PRINTTAB(X0,Y0+6)S$;S$;S$;V$;
  730 ENDPROC
  740 DEFPROC_PUT(PX,PY,C$)
  750 IF L%(PX,PY)=0 THEN COLOUR 128 ELS
E COLOUR 129+(L%(PX,PY)+1)/2
  760 PRINTTAB(PX,PY)C$;
  770 IF L%(PX,PY)=0 THEN L%(PX,PY)=T
  780 COLOUR128
  790 ENDPROC
  800 DEFPROC_TEST
  810 FOR J=4 TO 6
  820   PROC_CHECKX:PROC_CHECKY
  830 NEXTJ
  840 IF NOT FNF THEN 880
  850 FOR J=1 TO 3
  860   PROC_CHECKX:PROC_CHECKY
  870 NEXTJ
  880 IF NOT FNT THEN 930
  890 FOR J=1 TO 3
  900   PROC_CHECKXY
  910   PROC_CHECKX
  920 NEXTJ
  930 IF FNF OR FNT OR FNE OR FNE2 THEN
980
  940 FOR J=1 TO 3
  950   PROC_CHECKXY
  960   PROC_CHECKY
  970 NEXTJ
  980 ENDPROC
  990 DEFPROC_CHECKX
 1000 IF FNX(PS+R(J,1)) OR FNX(PS+R(J,2)
) THEN 1020
 1010 IF M%(PS+R(J,1),VP)=M%(PS+R(J,2),V
P) AND M%(PS+R(J,2),VP)=T THEN SC(T+1)=S
C(T+1)+1:PROC_SOUND
 1020 ENDPROC
 1030 DEFPROC_CHECKY
 1040 IF FNY(VP+R(J,1)) OR FNY(VP+R(J,2)
) THEN 1060
 1050 IF M%(PS,VP+R(J,1))=M%(PS,VP+R(J,2
)) AND M%(PS,VP+R(J,2))=T THEN SC(T+1)=S
C(T+1)+1:PROC_SOUND
 1060 ENDPROC
 1070 DEFPROC_CHECKXY
 1080 IF FNX(PS+R(J,1)) OR FNY(VP-R(J,1)
) OR FNX(PS+R(J,2)) OR FNY(VP-R(J,2)) TH
EN 1100
 1090 IF M%(PS+R(J,1),VP-R(J,1))=M%(PS+R
(J,2),VP-R(J,2)) AND M%(PS+R(J,2),VP-R(J
,2))=T THEN SC(T+1)=SC(T+1)+1:PROC_SOUND
 1100 ENDPROC
 1110 DEFPROC_SOUND
 1120 ENVELOPE1,1,-26,-36,-45,255,255,25
5,127,0,0,-127,126,0
 1130 SOUND1,1,100,50
 1140 ENDPROC
```