



The Game of Darts

The Game of Darts originated in England as a military pastime during the medieval period of the 1300's. Bored soldiers would compete by throwing spearheads and other sharp objects at upturned wine casks. This was encouraged so as to practice their aim and throwing skills.

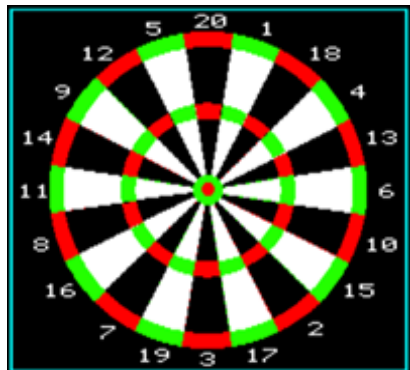
The natural structure of cross-sectioned tree trunks with their growth rings and radial cracks allowed competitors to further demonstrate their skills. The first purpose-made Darts were manufactured from solid wood, wrapped with a strip of lead for weight and fitted with flights made from split turkey feathers.

Various scoring systems have been used over the centuries, nevertheless it wasn't until the 1800's that Darts reached such a level of popularity a wave of innovations and developments fostered the beginnings of the modern numbering system.

The Modern Dart Board

All though there were many claimants to the inventor of the modern Dart Board, the consensus is given to a Lancashire carpenter named Brian Gamlin who in 1896 produced a Dart Board with the numbering scheme we know today.

At first view the numbering seems quite random, but is in fact a meticulously picked order, the design of which reduces the incidents of a 'lucky shot' and diminishes the element of chance, so that only skill and accuracy will attain the best scores.



Computer Game of Darts

Compared with other sports very few computer-based Darts Games have been released. The first notable one being Metronics 180 released in 1986 for the commodore 64, ZX Spectrum, Amstrad and Atari 8-bit. Indoor Sports released a game set in 1987 which included Darts and in 1991 Seta Corporation released Magic Darts!

In 2006 a version of a Darts Game was released for PlayStation 2 and PC. Fans were finally able to play a modern-day version based on the PDC World Championships. In 2011 Microsoft Studios added Darts to the Games included in their Kinect Sport sequel.

Darts is a pure sport with simple rules that belies the depth of tactical and technical ability required. While being a Professional shooting sport, darts is also a traditional Pub Game.

Arrows is a slang term for darts or darting, it refers to the thin stick weighted and pointed at one end with feathers at the other. Average is the score achieved after three arrows (darts) are thrown, which is a player's turn.

QBITS Darts

The most common Dart Board game is no doubt 301, where two people or teams compete to reduce a fixed score of 301 to zero. However, each player or side must start and end by throwing a double. This was my starting point, and then to add a 501 option. Being a little more ambitious I decided to add the clock face game. For this you throw a double for each number in sequence 1-20, then a 25 & 50 bullseye to finish.

QBITS Darts Intro

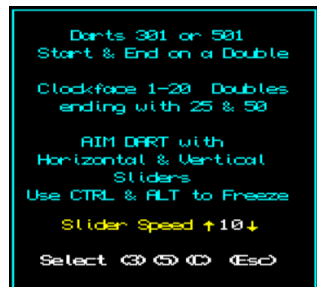
The Intro screen states a simple review of the options available and a means by which the player's choice can be made.



QBITS Darts Options

For the 301 and 501 options **Red** and **Green** teams, or individual players, can play against each other, the first to finish is the winner.

The Clock-face option is for a single player to complete in as few throws as possible.



QBITS Darts End of Game

At End of a Game the board is scrolled up with results displayed, and shows the number of Darts thrown.

QBITS Dartboard

The modern Dartboard is a disc with twenty segments, each divided into Single, Double, and Treble areas. The circumference being $2\pi r$ each segment is therefore $2\pi r/20$ or 18 degrees. I thought what could be simpler in writing the code for a Dartboard display, a circle divided into segments with sections for single, double and treble areas. When I first attempted this back in the eighties, I soon realised a little refreshment in tribometry wouldn't go amiss.

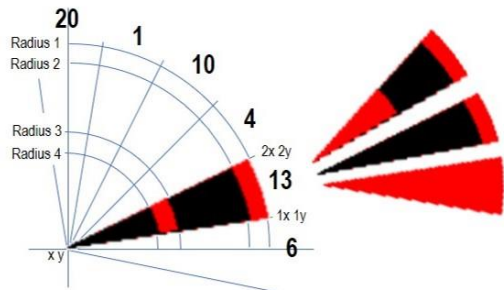
The centre of the circle gives the primary coordinates x, y, then each side of the segment require a straight LINE drawn from x, y TO 1x,1y and another from x, y TO 2x, 2y between then is drawn an ARC 1x,1y TO 2x,2y its radial angle applicable to the portion of the Dartboards circumference. That's the cleaver bit calculating the outer points to draw the LINE's and what is the angle of ARC.

QBITS Dartboard Segments

For a starting point the segment representing the number 6 is 9 degrees above and 9 degrees below the horizontal zero line. If we start at zero then add $\pi/20$ this provides an Offset to begin the drawing of a Dartboard segment. By adding a further $\pi/10$ (18 degrees) this give our second Offset. These are the angles, which with the length of radius and COS & SIN we can calculate 1x, 1y and 2x, 2y coordinates.

Using LINE and ARC to draw a segment then INK & FILL commands a coloured segment is drawn. Reducing the radius and change of colour this creates the Double, Treble and Single sections of a segment.

By adding multiples of $\pi/10$ to the angle, we can then process the next segment and so on around the board.



Last but not least we need a couple of FILL'd CIRCLES for the 25 and 50 at the centre.

QBITS Dartboard Numbering

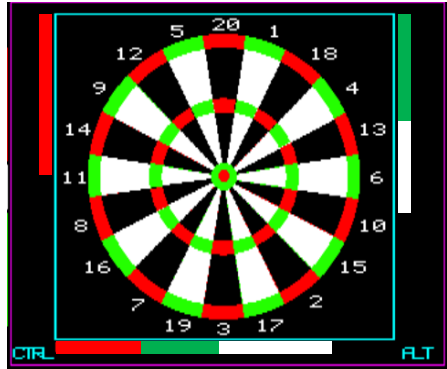
My 1985 QL ROM had Cursor Graphical Coordinate limited to Window#1, so for numbering my Dartboard I used the Pixel Coordinates system and fine-tuned it mostly by trial error. This built up an array of numbers and their x, y coordinates for use with the CURSOR command. Then it was a simple act to store them as DATA lines and use them in a FOR loop to display.



QBITS Dart Throws

In today's world a computer Game of Darts might use a motion detector to determine the accuracy of our throw. Back in the eighties this was not an option.

I don't know when or who introduced the Slider or Track bar as a graphical control in computing, but with a combination of Horizontal and Vertical Sliders a player can evaluate a cross point to aim their Dart.



The output from this gives a dx,dy coordinate for the Dart position with respect to the Dartboard centre coordinates x, y. The dart radius dr is calculated using Pythagoras theorem and the angle da with ACOS.

$$dr = \text{SQRT}((dy-y)^2 + (dx-x)^2) \quad \text{dr (dart radius)}$$

$$da = \text{ACOS}((dx-x)/dr) \quad \text{da (segment angle)}$$

Identifying the relevant segment number was achieved by taking the angle then adding the first Offset and dividing this by $\pi/10$. The only problem being the angle reduces once passed 180 degrees, to cater for this I add a π and subtract the angle from π . Using the INTeger of the segment it is then simply a FOR loop to read through a list until the right number is reached.

As for the **Double** and **Treble** or centre **Bullseye** circles these can be checked against the radius values set up for the dartboard.

```

1185 DEFine PROCedure dnum
1186 RESTORE 1185
1187 IF dy<50:da=PI+(PI-da)
1188 dt=INT(((da+PI/20)/(PI/10)))+1
1189 FOR seg=1 TO dt:READ num
1190 dm=1
1191 IF dr > 44 :dm=0
1192 IF dr<=44 AND dr>40:dm=2
1193 IF dr<=24 AND dr>20:dm=3
1194 IF dr<= 4 :num=25
1195 IF dr<1.7:num=50
1196 REMark Dartboard numbers / segment
1197 DATA 6,13,4,18,1,20,5,12,9,14,11,8,16,7,19,3,17,2,15,10,6
1198 END DEFine

```

calculates Dart positions in terms
of segment number 1 to 20
and dart multiplier
single
double
treble
or 25
or 50

Note: Adjustments can be made to the **Slider** Speed (sp) - see the opening lines of the following Program code.

QBITS Darts code

1000 REMark **QBITS_Darts_bas** [QBITS Darts 2023 Review - QPC2]

1002 dev\$='win2_':MODE 8:gx=0:gy=0 :REMark basic settings

1004 **WHEN ERROr :CONTINUE:END WHEN**

1006 REMark import QBITSConfig settings - QPC2:

1007 OPEN _IN#9,dev\$&'QBITSConfig':INPUT#9,gx\gy\dn\$:CLOSE#9

1010 Init_win:sp=10:Darts_Intro:QBITS_Darts

1012 **DEFine PROCEDURE Init_win**

1013 WINDOW#2,512,224,gx,gy :PAPER#2,0:CLS#2:BORDER#2,1,3 :SCALE#2,100,0,0

1014 WINDOW#0,512,32,gx,gy+224 :PAPER#0,0:CLS#0:BORDER#0,1,3

1015 ch=6:OPEN#ch,scr_:WINDOW#ch,144,26,gx+4,gy+100:BORDER#ch,1,7:INK#ch,7

1016 ch=5:OPEN#ch,scr_:WINDOW#ch,144,66,gx+4,gy+ 32:BORDER#ch,1,2:INK#ch,2

1017 ch=4:OPEN#ch,scr_:WINDOW#ch,144,66,gx+4,gy+128:BORDER#ch,1,4:INK#ch,4

1018 ch=3:OPEN#ch,scr_:WINDOW#ch,358,210,gx+152,gy+6

1019 WINDOW#1,280,190,gx+186,gy+12:PAPER#1,0:CLS#1:BORDER#1,1,5:SCALE#1,100,0,0

1020 CSIZE#2,2,1:OVER#2,1:**Arrow 2,6,14,7:Arrow 2,6,38,7**

1021 INK#2,2:FOR i=0 TO 1:CUSOR#2,4+i,8:PRINT#2,'QBITS DARTS'

1022 INK#2,6:FOR i=0 TO 1:CUSOR#2,6+i,9:PRINT#2,'QBITS DARTS'

1023 OVER#2,0:INK#2,3:LINE#2,50,2 TO 50,98 TO 168,98 TO 168,2 TO 50,2

1024 CSIZE#2,2,0:**QBold 2,5,0,8,144,204,'CTRL':QBold 2,5,0,8,460,204,'ALT'**

1025 **END DEFine**

1027 **DEFine PROCEDURE Arrow(ch,col,x,y)**

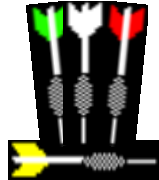
1028 INK#ch,col:FILL#ch,1:LINE#ch,x-6,y TO x-8,y+2 TO x-11,y+2

1029 LINE#ch TO x-9,y TO x-11,y-2 TO x-8,y-2 TO x-6,y:FILL#ch,0

1030 INK#ch,7 :FILL#ch,1:LINE#ch,x-11,y+.2 TO x+9,y TO x-11,y-.2:FILL#ch,0

1031 INK#ch,248:FILL#ch,1:CIRCLE#ch,x+2,y,3,.3,PI/2:FILL#ch,0

1032 **END DEFine**



1034 **DEFine PROCEDURE QBold(ch,col,d,w,x,y,str\$)**

1035 OVER#ch,1:INK#ch,col:sl=LEN(str\$)

1036 FOR a=1 TO sl

1037 FOR b=0 TO d:CUSOR#ch,b+x+a*w,y:PRINT#ch,str\$(a)

1038 END FOR a:OVER#ch,0

1039 **END DEFine**

1042 **DEFine PROCEDURE Clockface**

1043 ch=3:CSIZE#ch,2,0:INK#ch,2:FOR cn=1 TO 20:clk(cn)=cn

1044 n\$=25:**QBold ch,2,1,9,-9,176,n\$:QBold ch,2,1,9,317,176,n\$**

1045 FOR cn=1 TO 10:n\$=cn+10 **:QBold ch,2,1,9,317,cn*16,n\$**

1046 FOR cn=1 TO 9:n\$=cn **:QBold ch,2,1,9,0 ,cn*16,n\$**

1047 n\$=10 **:QBold ch,2,1,9,-9,160,n\$:cn=1**

1048 CSIZE#6,2,1:CUSOR#6,2,3:PRINT#6,'CLOCK':Arrow 2,7,38,49

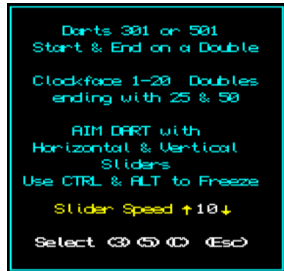
1049 **END DEFine**



```

1051 DEFINE PROCEDURE Darts_Intro
1052 DIM str$(10,35),dxy(6),clk(29):ch=1:CLS#ch:CSIZE#ch,2,0:INK#ch,7
1053 str$( 1)=" Darts 301 or 501"
1054 str$( 2)=" Start & End on a Double"
1055 str$( 4)=" Clockface 1-20 Doubles"
1056 str$( 5)=" ending with 25 & 50"
1057 str$( 7)=" AIM DART with"
1058 str$( 8)=" Horizontal & Vertical"
1059 str$( 9)=" Sliders"
1060 str$(10)="Use CTRL & ALT to Freeze"
1061 FOR lp=1 TO 10:QBold 1,5,0,10,0,lp*12,str$(lp)
1062 QBold 1,6,0,10,32,140,"Slider Speed ↑ ↓"
1063 QBold 1,7,1,10,12,164,"Select (3)(5)(C) (Q)uit":CURSOR#2,0,0
1064 REPEAT key
1065 CURSOR#1,186,140:PRINT#1,FILL$( ' ',2-LEN(sp))&sp : k=CODE(INKEY$(-1)):pcol=70
1066 SELECT ON k
1067 =208:sp=sp+1:IF sp>15:sp=15
1068 =216:sp=sp-1:IF sp< 5:sp= 5
1069 =51:score1=301:score=301:score2=301:EXIT key
1070 =53:score1=501:score=501:score2=501:EXIT key
1071 =67, 99:pcol=7:EXIT key
1072 =81,113:LRUN dn$:STOP
1073 =27:MODE 4:CSIZE#2,0,0:CSIZE#0,0,0:CLS#0:PRINT#0,'Bye...':STOP
1074 END SELECT
1075 END REPEAT key
1076 CLS#1:dartbd:bdcnt:bdnums
1077 IF pcol=7
1078 Clockface
1079 ELSE
1080 ch=5:CSIZE#ch,2,1:CURSOR#ch,6,2:PRINT#ch,'RED Team' :Arrow 2,2,38,71
1081 ch=4:CSIZE#ch,2,1:CURSOR#ch,6,2:PRINT#ch,'GREEN Team':Arrow 2,4,38,27
1082 pcol=0:teamsr:pcol=2
1083 END IF
1084 ch=1:INK#2,7
1085 END DEFINE

```



```

1087 DEFINE PROCEDURE Game_End
1088 BEEP 20000,1,10,300,30:CSIZE#1,2,1:PAUSE 50
1089 BLOCK#3,32,194,1,2,0 :BLOCK#3,36,194,314,2,0
1090 BLOCK#3,280,8,34,196,0:CLS#6:CLS#4:CLS#5
1091 ch=1:FOR up=1 TO 50:SCROLL#ch,-4:PAUSE 1
1092 IF pcol=2:shots=shot1:mes$='Winning Team':win$=' REDS '
1093 IF pcol=4:shots=shot2:mes$='Winning Team':win$=' GREENS '
1094 IF pcol=7:shots=shot3:mes$=' Clock-Face ':win$='Complete'
1095 QBold 1,pcol,1,12,54,60,mes$:QBold 1,pcol,1,14,68,90,win$
1096 QBold 1,7,0,12,32,130,'Darts Thrown: '&shots:PAUSE:Darts_Intro
1097 END DEFINE

```



```

1099 DEFine PROCEDURE QBITS_Darts
1100 dp1=0:shot1=0:dp2=0:shot2=0:dp3=0:shot3=0
1101 REPEAT Darts
1102 FOR p=1 TO 6 STEP 2
1103   sliders:dx(p)=dx:dx(p+1)=dy
1104   IF pcol=2
1105     dp1=num*dm:shot1=shot1+1
1106     IF score1=score AND dm<>2:dp1=0
1107     IF score1-dp1=0 AND dm=2:Game_End:RETURN
1108     IF score1-dp1<=1 OR score1<dp1:dp1=0:EXIT p
1109     score1=score1-dp1:teamschr
1110   END IF
1111   IF pcol=4
1112     dp2=num*dm:shot2=shot2+1
1113     IF score2=score AND dm<>2:dp2=0
1114     IF score2-dp2=0 AND dm=2:Game_End:RETURN
1115     IF score2-dp2<=1 OR score2<dp2:dp2=0:EXIT p
1116     score2=score2-dp2:teamschr
1117   END IF
1118   IF pcol=7
1119     IF cn<21 AND dm=2 AND num=clk(cn)
1120       IF cn<10:n$=cn:QBold 3,7,1,9, 0,cn*16,n$
1121       IF cn=10:n$=cn:QBold 3,7,1,9, -9,cn*16,n$
1122       IF cn>10:n$=cn:QBold 3,7,1,9,317,(cn-10)*16,n$
1123       cn=cn+1:PAUSE 20:BEEP 2000,5,10
1124     END IF
1125     IF cn=21 AND num=25:cn=cn+1:n$=25:QBold 3,7,1,9, -9,176,n$
1126     IF cn=22 AND num=50:cn=cn+1:n$=50:QBold 3,7,1,9,317,176,n$
1127     shot3=shot3+1:IF cn=23:Game_End:RETURN
1128   END IF
1129 END FOR p
1130 ch=1:PAUSE 20
1131 FOR n=1 TO 6 STEP 2
1132   dx=dx(n):dy=dx(n+1):dc=0:dart
1133 END FOR n
1134 dartbd:bdcnt:bdnums:dp1=0:dp2=0:dp3=0:IF pcol<>7:pcol=6-pcol:ELSE pcol=7
1135 END REPEAT Darts
1136 END DEFine

```



```

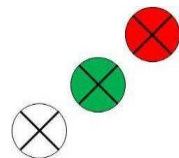
1138 DEFine PROCEDURE teamschr
1139 ch=5:INK#ch,2:CUSOR#ch,6,24:PRINT#ch,score1;" "
1140 IF pcol=2:CUSOR#ch,p*18,44 :PRINT#ch,FILL$(' ',2-LEN(dp1))&dp1:CLS#ch,4
1141 ch=4:INK#ch,4:CUSOR#ch,6,24:PRINT#ch,score2;" "
1142 IF pcol=4:CUSOR#ch,p*18,44 :PRINT#ch,FILL$(' ',2-LEN(dp2))&dp2:CLS#ch,4
1143 END DEFine

```

```

1145 DEFine PROCEDURE dart
1146 IF dc>0:BEEP 1000,1,200,60,50,60
1147 ch=1:INK#ch,dc
1148 FILL#ch,1:CIRCLE#ch,dx,dy,2.5:FILL#ch,0
1149 INK#ch,0:LINE#ch,dx-2,dy-2 TO dx+2,dy+2
1150 LINE#ch,dx-2,dy+2 TO dx+2,dy-2
1151 END DEFine

```

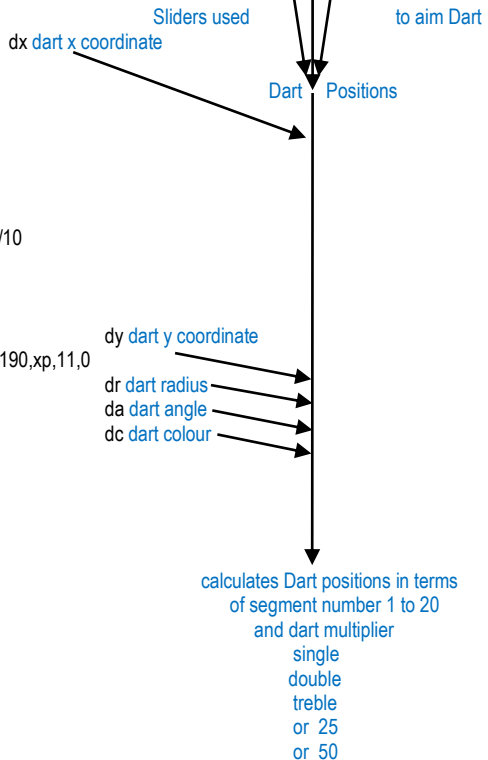
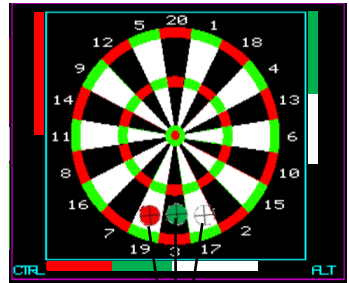


```

1153 DEFINE PROCEDURE sliders
1154 ch=2:BLOCK#ch,8,190,174,11,0:BLOCK#ch,8,190,466,11,0
1155 BLOCK#ch,278,6,185,202,0
1156 REPEAT lp_x
1157 FOR h=0 TO 278 STEP 4
1158   BLOCK#ch,h,6,185,202,pcol:PAUSE sp/10
1159   IF KEYROW(7)=2:EXIT lp_x
1160 END FOR h
1161 FOR h=278 TO 0 STEP -4
1162   BLOCK#ch,278-h,6,h+185,202,0:PAUSE sp/10
1163   IF KEYROW(7)=2:EXIT lp_x
1164 END FOR h
1165 END REPEAT lp_x
1166 dx=(h*.383)
1167 IF pcol=2:xp=174:ELSE xp=466
1168 REPEAT lp_y
1169 FOR v=4 TO 190 STEP 2
1170   BLOCK#ch,8,v,xp,11,pcol:PAUSE sp/10
1171   IF KEYROW(7)=4:EXIT lp_y
1172 END FOR v
1173 FOR v=180 TO 4 STEP -2
1174   BLOCK#ch,8,190-v,xp,11+v,0:PAUSE sp/10
1175   IF KEYROW(7)=4:EXIT lp_y
1176 END FOR v
1177 END REPEAT lp_y
1178 dy=100-(v*.521)
1179 BLOCK#ch,278,6,185,202,0:BLOCK#ch,8,190,xp,11,0
1180 dr=SQRT((dy-50)^2+(dx-54)^2)
1181 da=ACOS((dx-54)/dr)
1182 dc=pcol:dart:dnum
1183 END DEFINE

1185 DEFINE PROCEDURE dnum
1186 RESTORE 1185
1187 IF dy<50:da=PI+(PI-da)
1188 dt=INT((da+PI/20)/(PI/10))+1
1189 FOR seg=1 TO dt:READ num
1190 dm=1
1191 IF dr > 44 :dm=0
1192 IF dr<=44 AND dr>40:dm=2
1193 IF dr<=24 AND dr>20:dm=3
1194 IF dr<= 4 :num=25
1195 IF dr<1.7:num=50
1196 REMark Dartboard numbers / segment
1197 DATA 6,13,4,18,1,20,5,12,9,14,11,8,16,7,19,3,17,2,15,10,6
1198 END DEFINE

```




```

1200 DEFINE PROCEDURE dartbd
1201 x=54:y=50:an=PI/20:dx=0:dy=0
1202 FOR f=1 TO 10
1203 c1=2:c2=0:anseg:bdseg
1204 c1=4:c2=7:anseg:bdseg
1205 END FOR f
1206 END DEFINE

```

```

1208 DEFINE PROCEDURE anseg
1209 x1=COS(an):y1=SIN(an)
1210 an=an+PI/10
1211 x2=COS(an):y2=SIN(an)
1212 END DEFINE

```

```

1214 DEFINE PROCEDURE bdseg
1215 r=44:c=c1:dwseg
1216 r=40:c=c2:dwseg
1217 r=24:c=c1:dwseg
1218 r=20:c=c2:dwseg
1219 END DEFINE

```



```

1221 DEFINE PROCEDURE dwseg
1222 ch=1:FILL#ch,1:INK#ch,c
1223 ARC#ch,x+x1*r,y+y1*r TO x+x2*r,y+y2*r,PI/10
1224 LINE#ch TO x,y TO x+x1*r,y+y1*r:FILL#ch,0
1225 END DEFINE

```

```

1227 DEFINE PROCEDURE bdcnt
1228 INK#ch,4:FILL#ch,1:CIRCLE#ch,x,y,4 :FILL#ch,0
1229 INK#ch,2:FILL#ch,1:CIRCLE#ch,x,y,1.7:FILL#ch,0
1230 END DEFINE

```

```

1232 DEFINE PROCEDURE bdnums
1233 RESTORE 1139:ch=1:OVER#ch,1:CSIZE#ch,2,0:INK#ch,7
1234 FOR n=1 TO 20
1235 READ num,nx,ny:CURSOR#ch,nx,ny:PRINT#ch,num
1236 END FOR n
1237 OVER#ch,0 :REMark Board nums,x,y coordinates
1238 DATA 1,172,5,18,206,18,4,236,38,13,246,62,6,256,90
1239 DATA 10,246,118,15,230,142,2,206,162,17,162,176,3,130,178
1240 DATA 19,86,176,7,60,164,16,20,142,8,13,118,11,2,90
1241 DATA 14,4,62,9,28,38,12,46,18,5,92,5,20,126,1
1242 END DEFINE

```