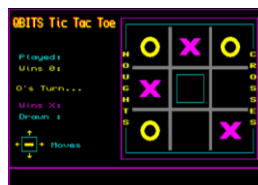


QBITS SuperBASIC Progs

GAMES One

Tic Tac Toe



Mine Detector



Tile Slider



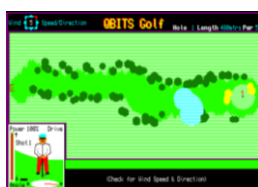
Conundrum

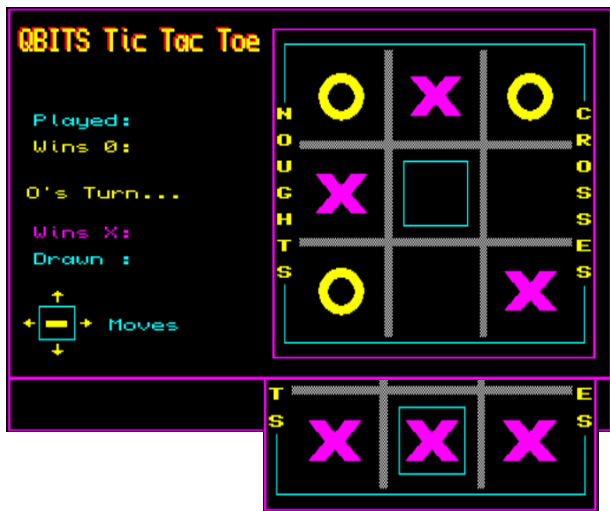


Darts



Golf





TIC TAC TOE Introduction

Described as a deceptively easy game the origins of Nought and Crosses is unknown, but some evidence suggests that something very similar was played by the Ancient Egyptians. Each game has three possible outcomes, a win by Noughts or Crosses or a draw, in which neither side wins.

QBITS TIC TAC TOE

This QBITS SuperBASIC Game presents a three-by-three Grid. A player chooses which side to play, and who goes first Noughts or Crosses is randomly selected by the Prog? Both players on average will get an even chance at going first, the choice of Player indicated with **O's** or **X's Turn...** Use the Cursor keys to select a grid position, then press the spacebar and the relevant Nought or Cross is drawn.

Trying to decide if the outcome will be a draw would require analysing multiple combinations. The variations and combinations give 255,168 possible ways of playing Tic Tac Toe. The coding for resultant moves was therefore limited to deciding the state of play for three of a kind either in a horizontal, vertical or diagonal row of cells across the grid.

QBITS TIC TAC TOE Strategy

If you are the first to go a simple **Strategy** is placing your **Nought** or **Cross** in any corner. The aim is to create three possible ways to complete a three in a row. This move will give your opponent the most opportunities to make a mistake and thereby give you the best chance of a win. However, if your opponent places their Nought or Cross in the centre, this will make it all the more harder.

Games with two equally matched players invariably ends in a draw (i.e. undecided). The game is too short for any initiative by the second player to force a win, they must rely on the first player making a mistake.

QBITS TIC TAC TOE Code

1000 REMark **QBITS_TTT_bas** [QBITS TIC TAC TOE 2024 Review – QPC2] vM19

1002 dev\$='win1_': MODE 8:gx=0:gy=0 :REMark Basic Settings

1004 **WHERN ERror** :CONTINUE:**END WHEN**

1006 REMark **Import QBITSConfig Settings – QPC2**

1007 OPEN _IN#9,'ram2_QBITSConfig':INPUT#9,gx\gy\dn\$:CLOSE#9

1010 **Init_win**:chk=0:**QBITS_TTT**

1012 **DEFine PROCEDURE Init_win**

[Create screen layout](#)

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

1014 WINDOW#1,274,200,gx+224,gy+12:PAPER#1,0:BORDER#1,1,3:CLS#1

1015 WINDOW#0,512, 32,gx,gy+224 :PAPER#0,0:BORDER#0,1,3:CLS#0

1016 CSIZE#2,2,1:OVER#2,1:SCALE#1,100,0,0

1017 INK#2,2:FOR i=0 TO 1:CUSOR#2,4+i,8:PRINT#2,'QBITS Tic Tac Toe'

1018 INK#2,6:FOR i=0 TO 1:CUSOR#2,6+i,9:PRINT#2,'QBITS Tic Tac Toe'

1019 CSIZE#2,2,0:Str1\$='NOUGHTS':Str2\$='CROSSES' :c=1:r=2

1020 FOR a=1 TO 7

1021 FOR b=0 TO 2:CUSOR#2,224+b,42+a*16:PRINT#2,Str1\$(a)

1022 FOR b=0 TO 2:CUSOR#2,478+b,42+a*16:PRINT#2,Str2\$(a)

1023 END FOR a

1024 OVER#2,0:m=0:OG=0:XG=0:INK#1,5:LINE#1,3,22 TO 3,4 TO 98, 4 TO 98,22

1025 **RESTORE 1021**:PG=0:FG=0:LINE#1,3,78 TO 3,96 TO 98,96 TO 98,78:INK#1,248

1026 FOR i=1 TO 8:**READ col,x,y,str\$**:INK#2,col:CUSOR#2,x,y:PRINT#2,str\$

1027 DATA 6,33,168,'↑',6,33,200,'↓',6,9,184,'←' →'

1028 DATA 5,82,186,'Moves',5,18,62,'Played:',6,18,78,'Wins 0:'

1029 DATA 3,18,130,'Wins X:',5,18,146,'Draw ':BLOCK#2,18,4,30,188,6

1030 FOR i=1 TO 4

1031 **READ x1,y1,x2,y2,x3,y3,x4,y4**

1032 FILL 1:LINE x1,y1 TO x2,y2 TO x3,y3 TO x4,y4 TO x1,y1:FILL 0

1033 END FOR i

1034 DATA 8,36,94,36,94,34,8,34, 8,66,94,66,94,64,8,64

1035 DATA 35,6,35,94,37,94,37,6, 65,6,65,94,67,94,67,6

1036 INK#2,5: LINE#2,8,10 TO 8,19 TO 18,19 TO 18,10 TO 8,10

1037 **END DEFine**

1039 **DEFine PROCEDURE QBITS_TTT**

[Game Control](#)

1040 **REPeat loop**

1041 IF chk=0:INK#2,5:CUSOR#2,12,106:PRINT#2,"(N)EW / (Q)uit"

1042 IF chk=1:INK#2,6:CUSOR#2,12,106:PRINT#2,"O's Turn... "

1043 IF chk=2:INK#2,3:CUSOR#2,12,106:PRINT#2,"X's Turn... "

1044 x=11+c*30:y=1*30:**Tile_Hgl x,y**:k=CODE(INKEY\$(-1)):**Tile_Hgl x,y**

[calculate x-y from c-r](#)

1045 **SElect ON k**

1046 =27:MODE 4:CSIZE#2,0,0:CSIZE#0,0,0:CLS#0:PRINT#0,'Bye...':STOP

1047 =81,113:IF chk=0:LRUN dn\$:ELSE chk=0 :REMark (Q)uit Game

1048 =77,110:**Tile_CLS**:chk=RND(1 TO 2) :REMark (N)ew Game

1049 =192:c=c-1:IF c<0:c=0 :REMark Left

1050 =200:c=c+1:IF c>2:c=2 :REMark Right

1051 =208:r=r+1:IF r>3:r=3 :REMark Up

1052 =216:r=r-1:IF r<1:r=1 :REMark Down

1053 = 32:**RESTORE 1076**:IF chk=1:**Nought**:ELSE IF chk=2:**Cross**

1054 **END SElect**

1055 **END REPeat loop**

1056 **END DEFine**

1058 REMark QBITS TIC TAC TOE Graphics

```
1060 DEFine PROCEDURE Tile_Hgl(x,y)
1061 INK 5:OVER -1:LINE x,y TO x+20,y TO x+20,y-20 TO x,y-20 TO x,y:OVER 0
1062 END DEFine
```



```
1064 DEFine PROCEDURE Nought
1065 IF Grid(c+1,r)=0:Grid(c+1,r)=79:chk=2:x=x+10:y=y-10:ELSE RETurn
1066 INK 6:FILL 1:CIRCLE x,y,7:FILL 0:INK 0:FILL 1:CIRCLE x,y,4:FILL 0
1067 BEEP 2000,5,10,0,0,0,0,0:Tile_Chk 79
1068 END DEFine
```



```
1070 DEFine PROCEDURE Cross
1071 IF Grid(c+1,r)=0:Grid(c+1,r)=88:chk=1:ELSE RETurn
1072 INK 3:x=x+10:y=y-10:FILL 1:LINE x-8,y+6 TO x-3,y+6 TO x+2,y TO x-3,y-6
1073 LINE TO x-8,y-6 TO x-3,y TO x-8,y+6:FILL 0:FILL 1:LINE x+8,y+6 TO x+3,y+6
1074 LINE TO x+2,y TO x+3,y-6 TO x+8,y-6 TO x+3,y TO x+8,y+6:FILL 0
1075 BEEP 2000,5,10,0,0,0,0,0:Tile_Chk 88
1076 END DEFine
```



1078 REMark QBITS TIC TAC TOE Game Checks

For three in a row, there are eight possible combinations. A FOR loop READ's the combinations as DATA sets and compares the three Grid entries. If they all test the same ie. all Noughts or all Crosses then a win is declared and the score is updated.

```
1080 DEFine PROCEDURE Tile_Chk(n)
1081 FOR i=1 TO 8
1082   READ a,b,c,r,d,e
1083   IF Grid(a,b)=n AND Grid(c,r)=n AND Grid(d,e)=n:Tile_Win:RETurn
1084 END FOR i
1085 m=m+1:IF m=9:m=0:n=0:Tile_Win
1086 DATA 1,1,1,2,1,3, 1,1,2,2,3,3, 1,1,2,1,3,1, 1,2,2,2,3,2
1087 DATA 3,3,3,2,3,1, 3,3,2,3,1,3, 1,3,2,2,3,1, 2,1,2,2,2,3
1088 END DEFine
```

1090 DEFine PROCEDURE Tile_Win Displays Updated Score

```
1091 chk=0 :PG=PG+1:INK#2,5:CUSOR#2,124, 64:PRINT#2,FILL$(' ',2-LEN(PG))&PG
1091 IF n=79:OG=OG+1:INK#2,6:CUSOR#2,124, 80:PRINT#2,FILL$(' ',2-LEN(OG))&OG
1093 IF n=88:XG=XG+1:INK#2,3:CUSOR#2,124,132:PRINT#2,FILL$(' ',2-LEN(XG))&XG
1094 IF n=0 :FG=FG+1:INK#2,5:CUSOR#2,124,148:PRINT#2,FILL$(' ',2-LEN(FG))&FG
1095 BEEP 5000,20,12,40,8,8,0,0
1096 END DEFine
```

```
1098 DEFine PROCEDURE Tile_CLS
1099 DIM Grid(3,3):m=0:n=0:c=1:r=2:RESTORE 1103:INK 0
1100 FOR i=1 TO 9
1101   READ x,y,x1,y1:FILL 1:LINE x,y TO x1,y TO x1,y1 TO x,y1 TO x,y:FILL 0
1102 END FOR i
1103 DATA 11,30,31,10, 41,30,61,10, 71,30,91,10, 11,60,31,40, 41,60,61,40
1104 DATA 71,60,91,40,11,90,31,70, 41,90,61,70,71,90,91,70
1105 END DEFine
```