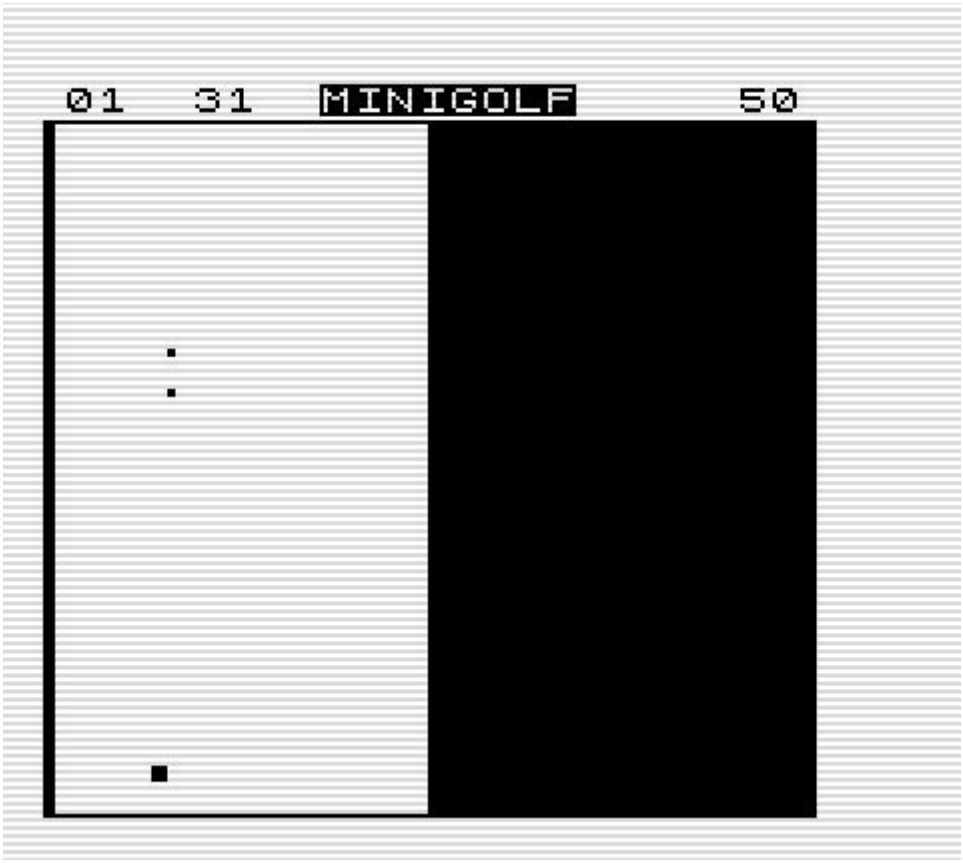
MiniGolf



This one has always been on my list to code. The challenge seemed to fit enough levels to have a nice game. 11 levels have made it to this game. My 50th game in 1K hires is once again a special game. Enough gameplay to keep you busy for hours.

```
; MINIGOLF, Game 50 in 1K Hires

; 11 levels of golf in 1K
; ==

; line1 .........
; line2 .       .
; line3 .   .   .
; line4 .   .....
; line5 .....   .
; line6 put, copy of 1..5 with put set
; line7 person in angle around ball, copy of 1..6 with pixel

maxlev     EQU  11
len        EQU  25
counter    EQU  hits+1
l2delay    EQU  #4000+len+1
notplayr   EQU  l2delay+4

? * TORNADO *

           ORG  #4009            ;#4009
           DUMP 49161

basic      LD   C,10             ; preset for delay copy
           JR   init0
```

```
           DEFB 236,212,28        ; The BASIC
           DEFB 126               ; fully placed over sysvar
           DEFB 143,0,18          ; start to BASIC=#4009

eline      DEFW last              ; needed by loading
chadd      DEFW last-1
xptr       DEFW 0
stkbot     DEFW last
stkend     DEFW last
berg       DEFB 0
mem        DEFW 0
           DEFB 128

init1      JP   init             ; init can be anywhere

; all above reusable AFTER loading

lastk      DEFB 255,255,255       ; used by ZX81
margin     DEFB 55                ; used by ZX81
nxtlin     DEFW basic             ; reusable after load

init0      XOR  A                 ; intruptcounter reset
           DEFB 254               ; CP n ; skip flagx
flagx      DEFB 0

           LD   B,A               ; BC now copy size
           DEFB 17                ; LD DE,nn  ; skip taddr

taddr      DEFW 3213              ; used by ZX81
           EX   AF,AF'
           DEFB 17                ; LD DE,nn  ; skip frames

frames     DEFW 65535             ; decreased by 1 by ZX81
hits       JR   init1             ; useable, here hitcounter
prcc       DEFB 188               ; used by ZX81
sposn      DEFB 33,24             ; used by ZX81
cdflag     DEFB 64                ; used by ZX81

; the hires display routine starts here
hr         LD   HL,lowres+#8000   ; the lowres display
           LD   BC,#211           ; minimum needed
           LD   A,#1E
           LD   I,A
           LD   A,#FB
           CALL #2B5
           LD   IX,nlin2

           LD   B,05              ; sync hires display
hr00       DJNZ hr00

xpos       LD   HL,#4000          ; preset erase ball
           LD   A,H
           LD   I,A
           LD   DE,dispdata       ; the pointers to each line
           LD   B,175             ; 175 lines

nlin2      LD   A,B
           AND  1
           JR   NZ,l2delay        ; second line needs timing

nline      INC  DE                ; point to next line
repair     LD   (HL),0            ; set line back, old value

           LD   A,B
```

```
ypos       CP   255              ; y pos of ball
           JR   NZ,notplayr      ; just show the line

           CP   (HL)             ; timing
xbit       LD   (HL),170         ; set ballpixel

npl        LD   A,(DE)           ; get linepointer
           DEC  B                ; decrease linecounter
           LD   R,A
           JP   NZ,lbuf+#8000    ; show remaining lines

exit       CALL #292             ; back from intrupt
           CALL #220
           LD   IX,hr
           JP   #2A4

; "start" of screenlines (also 1 line built at #4001)
line3      DEFB 1,192
           DEFW 0,0,0,0,0
line4      DEFB 1,192
           DEFW 0,0,0,0,0
line5      DEFB 1,255,255
           DEFB 255,255,255,255
           DEFB 255,255,255,255
           DEFB 255,255
           DEFB 192,0,0,0,0,0
           DEFB 0,0,0,0,0
line2      DEFB 1,192
           DEFW 0,0,0,0,0,0
           DEFW 0,0,0,0,0

; during game used as copybuffer for put
line6      DEFB 1                ; 01

l2delayc   PUSH HL               ; 02 sync display of each
           LD   A,(DE)           ; 03 second line
           DEC  HL               ; 04
           POP  HL               ; 05
           DEFB 24               ; 06 JR
           DEFB npl-notplayr-2   ; 07 sync timing not ball

line6in    LDIR                  ; 09 built a black screen
           LD   HL,l2delayc      ; 12
           LD   (hits),HL        ; 15 set HITS to no hiscore
           LD   DE,#4000+len+1   ; 18 reuse sysvar to save
           LD   C,6              ; 20 bytes for an exta level
           LDIR                  ; 22 copy over sysvar
           LD   HL,swapx         ; 25

line7      LD   DE,init1-2       ; 03 saves 1 byte in code
           LD   C,4              ; 05 so cosmetic screen is
           LDIR                  ; 07 again ok
           JP   put              ; 10

swapx      XOR  A                ; 11
           SUB  E                ; 12 vertical collision, so
           LD   E,A              ; 13 x-direction is altered
           RET                   ; 14

delay      LD   HL,frames        ; 17 delay goes over sysvar
           LD   A,(HL)           ; 18
           SUB  2                ; 20
wfr        CP   (HL)             ; 21
           JR   NZ,wfr           ; 23
```

```
           RET                    ; 24
           DEFB 0                 ; 25 line7 filler
; end of all possible screenlines

mkdydx     CP   D                 ; test against #4n
           CCF
           JR   NC,posy           ; nr is positive
           NEG                    ; make nr pos
posy       LD   L,A               ; l is abs(vector)
           SBC  A,A
           ADD  A,A
           INC  A
           RET                    ; a is dydx -1 or 1

makeline   LD   DE,dispdata+88    ; the end of screen
fcline     DEC  DE
           DJNZ fcline            ; find needed screenline

           LD   A,(DE)            ; get background line
           JR   NZ,skipera        ; double used, flag is preset
           LD   (moveloop+1),DE
           LD   (restval+1),A
skipera    INC  A                 ; display is 1 more
           LD   L,A
           LD   A,C               ; copy line as background

           LD   (DE),A            ; set this background in data
           CP   line6*256/256-1
           JR   NZ,skipdoub       ; the cursor is just 1 line
           DEC  DE
           LD   (DE),A            ; double
skipdoub   INC  A
           LD   E,A               ; DE must get background
           LD   H,#40
           LD   D,H
           LD   C,len
           LDIR                   ; copy background
           RET

; table stored in LBUF, double use of LBUF, data and runcode
taby       DEFB 7,8,9
lbuf       DEFB 10,11,12,12,12,12,12
           DEFB 11,10,9
           DEFB 8,7,6,5,4,3
           DEFB 2,2
tabx       DEFB 2,2,2,3,4,5
           DEFB 6
outofrange JP   (IX)

plot       LD   A,B               ; get y
           ADD  A,A               ; double for display
           LD   (ypos+1),A        ; set y of ball in HR
           LD   A,#A7             ; "AND A" to reset carry
plotplay   LD   (flagt),A         ; set test to do
           PUSH BC                ; save x and y
           LD   HL,dispdata+88
fline      DEC  HL
           DJNZ fline             ; find fitting background
           LD   A,C               ; get x
           RRCA                   ; 4 pixels per byte
           RRCA                   ; so divide by 4
           AND  31                ; clear remainder of division
           ADD  A,(HL)            ; add full byte displacement
```

```
            LD   (setx+1),A        ; set xpos in hr

            LD   L,A
            LD   H,#40             ; hl points in screenline
            LD   A,(HL)            ; get screendata without ball
            LD   (rep+1),A         ; set in HR for clearing ba
            LD   A,C               ; get x
            AND  3                 ; remainder within byte
            INC  A                 ; always a number
            LD   B,A               ; 256 times = flicker
            LD   A,3               ; 2 pixels set
fbit        RRCA                   ; rotate pixels
            RRCA                   ; until correct position
            DJNZ fbit
            LD   C,A               ; save pixels
            XOR  (HL)              ; "add" background
flagt       NOP                    ; test what to do
            JR   C,explay          ; player different from ball
            LD   (xbit+1),A        ; set in HR the ball
            SUB  (HL)              ; subtract background
            CP   C                 ; collission test
            LD   A,(HL)            ; get old value
            POP  HL                ; get x and y
            JR   NZ,col            ; no match is collission
setx        LD   A,0               ; only when plotting the ball
            LD   (xpos+1),A
rep         LD   A,0
            LD   (repair+1),A
            LD   (dr5+1),HL
            JP   nxtlin            ; the delay routine

            LD   L,score*256/256
setnr       LD   H,score/256       ; all setnr have same highbyte
            LD   (HL),27           ; write number to screen
setten      INC  (HL)              ; 2 positions on screen
            SUB  10
            JR   NC,setten         ; calculate 0. - 9.
            ADD  A,38
            INC  HL
            LD   (HL),A            ; calculate .0 - .9
; exit through swapy saves a byte, no D-reg function here

swapy       SUB  D                 ; alter y-direction
            LD   D,A
            RET

explay      POP  BC                ; no collisiontest, move is ok
            RET

col         INC  A                 ; increase background byte
            JR   Z,swapy           ; horizontal line
            SUB  %00111101
            JP   NZ,init1-2

put         LD   HL,levnr+1        ; go to next round
            INC  (HL)

            LD   A,(HL)
            SUB  maxlev+1
            JR   NZ,nlevel         ; not at end of rounds?

restart     LD   (HL),1            ; reset levelcounter
            LD   HL,hits
```

```
           LD   A,(HL)            ; get nr of hits
           LD   (HL),0            ; reset all hits
high       CP   50                ; test against current hiscore
           JR   NC,start

           LD   (high+1),A        ; save new hiscore
           LD   L,hiscore*256/256
           CALL setnr             ; display hiscore

start      LD   A,(lastk)         ; game over, wait for
           SUB  %10111111         ; newline
           JR   NZ,start

           CALL setnr-2           ; reset score

nlevel     LD   HL,level1-4       ; start of leveldata
levnr      LD   C,maxlev          ; level counter

findlev    CALL #7B8              ; 4x INC HL in ROM, -1 byte
           LD   A,86              ; 88 lines per level, 2 fixed
           LD   DE,dispdata+1     ; the displaydata
set0       SUB  (HL)              ; subtract current nr lines
           PUSH AF                ; save lines
           LD   B,(HL)            ; get linerepeater
           INC  HL
setlevel   LD   A,C               ; test set or find level
           DEC  A
           JR   NZ,skip
           LD   (xpos+1),A        ; erase old put
           LD   A,(HL)            ; get line
           LD   (DE),A            ; set line
skip       INC  DE
           DJNZ setlevel          ; do all lines
           POP  AF                ; retrieve remaining lines
           INC  HL
           JR   NZ,set0           ; test screen end reached
           DEC  C
           JR   NZ,findlev        ; test right level reached

; HL now points to puthole
           LD   B,(HL)            ; get y puthole
           LD   C,line6*256/256-1 ; we must built line6
           INC  HL                ; point to x puthole
           PUSH HL                ; save for later

           OR   H                 ; set NZ flag
           CALL makeline          ; built line6

           POP  HL
           ADD  A,(HL)            ; get dx
           LD   E,A
           LD   A,%00111100       ; the graphical put
           LD   (DE),A            ; set put

           CALL #1300             ; ROM: DE from (HL), -1 byte
           LD   (dr5+1),DE        ; set ball xy from leveldata

stopball   LD   A,20              ; reset speed
           LD   (counter),A

; make line 7, then plot line
rotate     LD   A,B
           LD   HL,(dr5+1)        ; get xy ball
           LD   SP,#4400          ; drop PUSHes from collission
```

```
            PUSH HL
pointer     ADD  A,tabx*256/256

            SUB  taby*256/256
            CP   outofrange*256/256-taby*256/256
            JR   C,okp              ; not out of table
            SUB  28                 ; stay in table
okp         ADD  A,taby*256/256
            LD   (pointer+1),A
            LD   E,A                ; point to table entry
            LD   D,taby/256

            LD   A,(DE)             ; get dx+7
            SUB  7                  ; make it dx

            ADD  A,L                ; add x ball
            LD   C,A                ; C now x vector
            SUB  L
            CALL mkdydx             ; make dx and x-vector
            LD   (dydx+1),A         ; set dx

            LD   A,E
            SUB  tabx-taby          ; from dx table to dy table
            CP   taby*256/256       ; out of range test
            JR   NC,inrange
            ADD  A,28               ; go to end of table
inrange     LD   E,A

            LD   A,(DE)             ; get dy+7
            SUB  7                  ; make it dy

            ADD  A,H                ; add y ball
            LD   B,A                ; BC now vector position

            SUB  H
            LD   H,L                ; save x-vector
            CALL mkdydx             ; make dy and y-vector
            LD   (dydx+2),A         ; save dy
            LD   (vector+1),HL



            PUSH BC
            LD   C,line7*256/256-1 ; built man line
            XOR  A                  ; set Z flag
            CALL makeline
            POP  BC

            LD   A,55               ; "SCF" to set carry flag
            CALL plotplay
            LD   (HL),A             ; plot player

            POP  BC                 ; ball xy
            CALL plot               ; built plot in HR for ball

moveloop    LD   HL,0               ; erase player
restval     LD   (HL),0

            LD   HL,lastk
            LD   A,(HL)
            LD   B,27               ; rotate anticlockwise
            CP   %11111101
            JR   Z,rotate
```

```
            LD   B,1               ; rotate clockwise
            CP   %11111011
            JR   Z,rotate
            DEC  B
            CP   %11011111         ; test release
            JR   NZ,rotate         ; built screen, no move

w4up        CP   (HL)              ; test same key released
            JR   Z,w4up

power       LD   HL,counter        ; speedcounter

            CP   %11111011         ; Q-T ?
            JR   Z,inchl
            CP   %11111101         ; A-G ?
            JR   NZ,skipspeed      ; other key do nothing

dechl       DEC  (HL)
            JR   NZ,skipspeed
inchl       INC  (HL)

skipspeed   LD   A,(HL)
            CP   100
            JR   NC,dechl          ; stay in range

            LD   L,speed*256/256   ; show speed
            CALL setnr

            CALL nxtlin            ; delay loop on sysvar

            LD   A,(lastk)         ; read next key
            CP   %11011111
            JR   NZ,power          ; while not shot, read speed

            LD   HL,hits           ; we hit the ball
            INC  (HL)
            LD   A,(HL)
            CALL setnr-2           ; show nr of hits

dydx        LD   DE,0              ; dy and dx
vector      LD   HL,#0000          ; vector

dr5         LD   BC,0              ; plot pos oud
            XOR  A
            CP   L                 ; test dy=0
            JR   Z,noty

            LD   A,B               ; get current y
            SUB  D                 ; "add" displacement
            LD   B,A               ; set new y
            DEC  L                 ; 1 vertical step less

noty        XOR  A
            CP   H                 ; test dx=0
            JR   Z,notx

            LD   A,C               ; get current x
            SUB  E                 ; "add" displacement
            LD   C,A               ; set new x
            DEC  H                 ; 1 horizontal step less

notx        PUSH HL                ; save remaining vector
            CALL plot              ; plot the moving ball
            LD   HL,(dr5+1)        ; get plot position
```

```
            LD   A,H               ; test nearby wall
            SUB  7                 ; if so, skip decrease
            CP   79-3              ; to prevent stop
            JR   NC,skipdec        ; movement nearby wall

            LD   A,L               ; same test horizontally
            CP   99                ; but only right, left
            JR   NC,skipdec        ; is done by thick wall

            LD   HL,counter        ; stepcounter
            DEC  (HL)
            JP   Z,stopball        ; also repair SP

skipdec     POP  HL                ; get remaining vector
            LD   A,H
            OR   L
            JR   NZ,dr5            ; do all steps with vector
            JR   vector            ; get vector again

level1      DEFB 86,line4*256/256-1
            DEFB 6,4
            DEFB 40,80

level2      DEFB 66,line4*256/256-1
            DEFB 20,line2*256/256-1
            DEFB 10,23
            DEFB 30,80

level3      DEFB 66,line3*256/256-1
            DEFB 20,line2*256/256-1
            DEFB 80,19
            DEFB 30,80

level4      DEFB 39,line5*256/256-1
            DEFB 8,line2*256/256-1
            DEFB 39,line4*256/256-1
            DEFB 6,4
            DEFB 60,80

level5      DEFB 20,line2*256/256-1
            DEFB 46,line5*256/256-1
            DEFB 20,line2*256/256-1
            DEFB 76,2
            DEFB 20,14

level6      DEFB 12,line2*256/256-1
            DEFB 27,line3*256/256-1
            DEFB 8,line2*256/256-1
            DEFB 27,line3*256/256-1
            DEFB 12,line2*256/256-1
            DEFB 70,10
            DEFB 70,20

level7      DEFB 28,line2*256/256-1
            DEFB 1,line4*256/256-1
            DEFB 28,line2*256/256-1
            DEFB 1,line5*256/256-1
            DEFB 28,line2*256/256-1
            DEFB 70,20
            DEFB 20,20

level8      DEFB 21,line2*256/256-1
            DEFB 21,line3*256/256-1
            DEFB 2,line5*256/256-1
```

```
            DEFB 21,line3*256/256-1
            DEFB 21,line2*256/256-1
            DEFB 36,6
            DEFB 30,60

level9      DEFB 24,line2*256/256-1
            DEFB 1,line4*256/256-1
            DEFB 24,line2*256/256-1
            DEFB 1,line5*256/256-1
            DEFB 24,line3*256/256-1
            DEFB 12,line2*256/256-1
            DEFB 30,6
            DEFB 98,80

level10     DEFB 12,line2*256/256-1
            DEFB 24,line3*256/256-1
            DEFB 1,line4*256/256-1
            DEFB 12,line2*256/256-1
            DEFB 1,line5*256/256-1
            DEFB 24,line3*256/256-1
            DEFB 12,line2*256/256-1
            DEFB 30,6
            DEFB 80,60

level11     DEFB 13,line2*256/256-1
            DEFB 12,line4*256/256-1
            DEFB 12,line2*256/256-1
            DEFB 12,line5*256/256-1
            DEFB 12,line2*256/256-1
            DEFB 12,line4*256/256-1
            DEFB 13,line2*256/256-1
            DEFB 6,23
            DEFB 98,81

x           EQU  101

lowres      DEFB 118
score       DEFB 33,28,0,0          ; 50 as indicator 50th game
speed       DEFB 33,28,0,0          ; 50 as indicator 50th game
            DEFB "M"+x,"I"+x,"N"+x,"I"+x
            DEFB "G"+x,"O"+x,"L"+x,"F"+x,0,0,0,0,0
hiscore     DEFB 33,28              ; 50 as indicator 50th game
            DEFB 118

; end of memory - min. stacksize - screensize - current location
space       EQU  #4400-30-88-$
            DEFS space              ; remaining room (0) for code

; initialization done on the screen and reused memory
dispdata    EQU  $
init        LD   IX,hr              ; Hires mode
            LD   HL,delay
            LD   DE,nxtlin
            LDIR                    ; copy delay routine on sysvar

            LD   HL,#4000+len       ; built dataline1 on sysvar
setline1    LD   (HL),255
            DEC  L
            JR   NZ,setline1
            INC  L
            LD   (HL),L             ; finish the line

            LD   SP,#4400
            LD   HL,dispdata        ; make a black screen
```

```
            LD   (HL),B
            LD   DE,dispdata+1     ; which will keep top and
            LD   C,87              ; bottom for every level
            JP   line6in

vars        DEFB 128
?
last        EQU  $
```