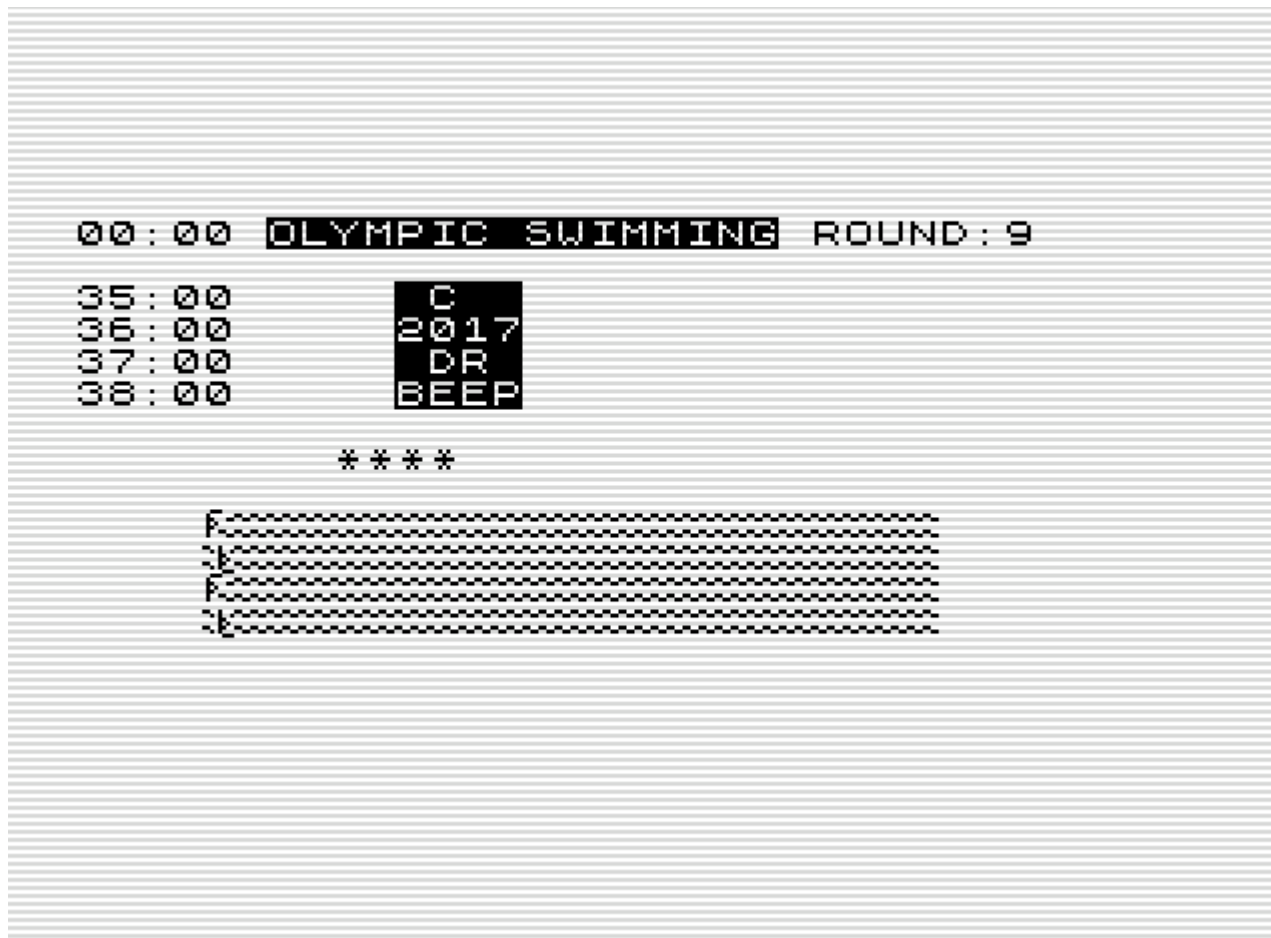


Olympic swimming



Why stop when you can continue coding? You all must get active. Do some sport. Swimming is also very good for obese persons. So no excuses, start sporting!

```
; Olympic swimming
```

```
? * TORNADO *
```

```
      ORG   #4009                ;#4009
      DUMP  49161
```

```
max      EQU   37
```

```
downwave EQU   vars
upwave    EQU   downwave+23
```

```
basic     LD    B,5                ; preset for 48K bug
           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

; 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                ; delay intrupts by
           DEFB 254             ; CP n ; skip flagx
flagx     DEFB 0

           EX  AF,AF'           ; intruptcounter reset
           DEFB 17              ; LD DE,nn ; skip taddr

taddr     DEFW 3213             ; used by ZX81
           JR  init1            ; continue to REAL init

frames    DEFW 65535            ; used by ZX81
coords    DEFB 0,0             ; useable
prcc      DEFB 188              ; used by ZX81
sposn     DEFB 33,24            ; used by ZX81
cdflag    DEFB 64              ; used by ZX81

lbuf      LD  R,A
           DEFW 0,0,0,0,0
           DEFW 0,0,0,0,0,0
           DEFB 0
           LD  L,C              ; get position udg
           LD  (HL),D           ; original background back
           JP  NZ,low           ; 48K bug
           JP  exit             ; back to low memory

hr         LD  HL,lowres+#8000  ; the lowres display
           LD  BC,#A21          ; minimum needed
           LD  A,#1E
           LD  I,A
           LD  A,#FB
           CALL #2B5

           LD  IX,lbuf+#8000    ; 8 tstates for JUMP

hr00       LD  B,07             ; sync hires display
           DJNZ hr00
           LD  A,(HL)

           LD  B,32              ; 4 players, 8 lines
           LD  H,#43
           LD  A,H
           LD  I,A
           LD  E,udgdata*256/256-32

low        LD  D,udgdata/256    ; repair D-reg from other use
           LD  L,B              ; each line has a pointer
           LD  A,(DE)           ; get xpos
           ADD A,(HL)           ; add linepointer
           LD  L,A              ; point to setpositionom
           INC E                ; point to graphic
           LD  A,(DE)           ; get graphic
           INC E                ; point to next graphicitem

```

```

        LD    C,L                ; save destination
        LD    D,(HL)            ; save original data
        LD    (HL),A            ; write graphic
        LD    L,B
        LD    A,(HL)            ; refetch linepointer
        DEC   B                ; decrease lines to show
        JP    (IX)              ; do display

exit    LD    B,200              ; screenfiller
ex2     EX    (SP),HL
        DJNZ  ex2

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

gameover POP   HL
        LD    HL,disq           ; disqualified text
winner  LD    DE,textline
        LD    BC,13
        LDIR

w4nl    LD    A,(lastk)         ; game over, wait for
        SUB   %10111111        ; newline
        JR    NZ,w4nl

start   LD    HL,textline
        LD    B,4
name    LD    (HL),23          ; "*" on screen
        INC   HL
        DJNZ  name

        LD    (HL),118          ; 2 newlines
        INC   HL
        LD    (HL),118

name4   LD    B,4
        LD    HL,textline
namepoint PUSH  BC
        RES   7,(HL)           ; show current cursorfield
wup     LD    A,(lastk)
        INC   A                ; wait for no key pressed
        JR    NZ,wup

wkey    LD    BC,(lastk)        ; wait for key pressed
        LD    A,C
        INC   A
        JR    Z,wkey

        CALL  readasci         ; translate key
        POP   BC
        DJNZ  namepoint        ; do next position
        JR    name4            ; restart

nlpress LD    SP,#4400          ; clear stacked values
        LD    HL,textline
        LD    DE,init0         ; some room for a name
        LD    BC,4             ; save name for hiscore
        LDIR

        LD    A,28              ; reset round counter
        LD    (roundnr),A

```

```

keyup      LD    A,(lastk)          ; again wait for no key
           INC   A
           JR    NZ,keyup

           LD    B,4
           LD    HL,#4000+20
resplayer  LD    (HL),0             ; all players on startline
           INC   HL
           DJNZ  resplayer

           CALL  makescr            ; built startscreen

           LD    HL,roundnr
           INC   (HL)               ; next round
           LD    A,(HL)
           CP    max
           LD    HL,wintext
           JR    Z,winner           ; champion on round 9

           LD    HL,onyrmarks
           LD    C,39               ; 4 textloops, 42 counts
           JR    quickstart         ; skip a delay

starter    PUSH  HL                ; save textindex
           LD    HL,frames          ; timer how long
           LD    A,(HL)             ; text stays on screen
           SUB   120
           LD    B,A               ; endvalue to B

wst        XOR   A                ; read full keyboard
           IN    A,(254)
           CPL
           AND   31                ; keypressed is
           JP    NZ,gameover        ; false start=game over

           LD    A,(HL)             ; get timer
           CP    B                 ; test end reached
           JR    NZ,wst

           POP   HL                ; retrieve textindex

quickstart LD    DE,textline
jrldi      LD    A,(HL)            ; get copied character
           LDI
           CP    118               ; test against Newline
           JR    NZ,jrldi          ; copy the text
           LDI                    ; copy second newline
           DEC   C
           JR    NZ,starter

clrtime    LD    HL,time           ; reset time
skip       LD    (HL),28
           INC   HL
           LD    A,(HL)
           CP    14                ; test ":"
           JR    Z,skip            ; skip ":"
           JR    NC,clrtime        ; stop at space

           LD    A,(frames)         ; reset time
           LD    (timecount+1),A

loop       LD    HL,frames
timecount  LD    A,0
           SUB   (HL)              ; calculate passed time

```

```

LD    B,A
LD    A,(HL)
CALL  NZ,disptime      ; adjust passed time
CALL  makescr          ; built the screen

LD    B,4              ; 4 players
LD    HL,#4000+20      ; xpos players
playmove DJNZ compmove
LD    A,1
movetest XOR 1
LD    (movetest+1),A    ; 1/2 playermove
JR    Z,nomove
LD    A,%01000000      ; all rows but H-NL
IN    A,(254)
CPL
AND    31
JR    Z,nomove          ; no key pressed
prevmove CP 0           ; other key then previous?
LD    (prevmove+1),A
JR    Z,nomove
domove INC (HL)         ; do a move
JR    nomove

compmove LD A,1         ; speed increases each round
DEC    A
JR    NZ,nomove
rseed  PUSH HL
LD    DE,0
LD    HL,(frames)
ADD    HL,DE
DEC    HL
LD    A,H
AND    #1F
LD    H,A
LD    (rseed+1),HL
LD    A,(HL)
POP    HL
CP    128               ; not always a move
JR    C,domove

nomove LD A,(HL)
SUB    46
JR    Z,whowon          ; someone reached finish
INC    HL
XOR    A
CP    B
JR    NZ,playmove       ; move all swimmers

LD    HL,compmove+1
DEC    (HL)
JR    NZ,noreset        ; next loop for computermove
LD    A,(roundnr)
LD    E,A
LD    A,max+1
SUB    E
LD    (HL),A

noreset LD HL,frames    ; some delay or the game
LD    A,(HL)            ; is too fast
SUB    4
delay  CP (HL)
JR    NZ,delay
JR    loop

```

```

whowon      CP    B
            JP    NZ,gameover      ; computer won
; test timetable
            LD    DE, timetable
            LD    B, 4
nexttab     LD    HL, time
            PUSH  DE
            LD    C, 5
hitest      LD    A, (DE)
            CP    (HL)
            INC   HL
            INC   DE
            JR    Z, nbit
            JR    C, nexthi        ; test next in table

; we have a faster time, so shift table
            LD    DE, endtab
            LD    HL, lasttab-1
            XOR   A
            DEC   B
            JR    Z, noshift        ; last place in table
calcsz      ADD   A, 15             ; insert faster in table
            DJNZ  calcsz
            LD    C, A
            LDDR                     ; table now shifted
noshift     POP   DE               ; get place in table
            LD    HL, time         ; your time in this round
            LD    C, 5
            LD    A, C
            LDIR                     ; copy faster time
            LD    HL, init0        ; your name
            ADD   A, E
            LD    E, A             ; position of name in table
            LD    C, 4             ; name is 4 positions
            LDIR                     ; and copy the name

wait2       LD    HL, frames
            LD    A, (HL)
            SUB   100
delay2      CP    (HL)
            JR    NZ, delay2        ; 2 seconds to release keys

keyupjp     JP    keyup           ; do next round

wintext     DEFB  "C"+n, "H"+n, "A"+n, "M"+n, "P"+n, "I"+n,
            DEFB  "O"+n, "N"+n
            DEFB  118, 118

nbit        DEC   C
            JR    NZ, hitest
nexthi      POP   DE
            LD    A, E
            ADD   A, tab2-timetable
            LD    E, A
            DJNZ  nexttab
            JR    wait2            ; no hi, but next round

makescr     LD    HL, #4000+20     ; players x-pos
            LD    DE, udgdata-32   ; start of udg screen
udgs2scr    LD    B, H
            LD    A, (HL)          ; get X-pos
            AND   1                ; bit 0 needed for right udg
            ADD   A, A
            ADD   A, A

```

```

        INC    A
        ADD    A,A
        LD     C,A                ; BC now correct UDG-pointer
udg2scr  LD     A,(HL)             ; get X again
        RRCA                    ; half X needed
        AND    127
        LD     (DE),A            ; store X
        INC    DE
        LD     A,(BC)            ; get UDG
        LD     (DE),A
        LD     A,C
        INC    BC
        INC    DE
        AND    7
        DEC    A
        JR     NZ,udg2scr        ; test end of UDG
        INC    HL
        LD     A,L
        CP     24
        JR     NZ,udgs2scr       ; test end of all players
        RET

n        EQU   101
t        EQU   27
disq     DEFB  "D"+n,"I"+n,"S"+n,"Q"+n,"U"+n,"A"+n
        DEFB  "L"+n,"I"+n,"F"+n,"I"+n,"E"+n,"D"+n
        DEFB  118,118
onryrmarks DEFB "O"-t,"N"-t,0,"Y"-t,"O"-t,"U"-t,"R"-t
        DEFB  0,"M"-t,"A"-t,"R"-t,"K"-t,"S"-t,118,118
getset   DEFB  0,0,0
        DEFB  "G"-t,"E"-t,"T"-t,0,"S"-t,"E"-t,"T"-t,118,118
go        DEFW  0,0
        DEFB  0,"G"-t,"O"-t,118
lowres   DEFB  118
time     DEFB  28,28,14,28,28,0
        DEFB  "O"+n,"L"+n,"Y"+n,"M"+n,"P"+n,"I"+n,"C"+n,128
        DEFB  "S"+n,"W"+n,"I"+n,"M"+n,"M"+n,"I"+n,"N"+n
        DEFB  "G"+n,0,"R"-t,"O"-t,"U"-t,"N"-t,"D"-t,14
roundnr  DEFB  max
        DEFB  118
        DEFB  118

timetable DEFB  31,33,14,28,28,0,0,0,0,0
        DEFB  128,"C"+n,128,128,118

tab2     DEFB  31,34,14,28,28,0,0,0,0,0
        DEFB  158,156,157,163,118

seclast  DEFB  31,35,14,28,28,0,0,0,0,0
        DEFB  128,"D"+n,"R"+n,128,118

lasttab  DEFB  31,36,14,28,28,0,0,0,0,0
        DEFB  "B"+n,"E"+n,"E"+n,"P"+n
endtab   DEFB  118
        DEFB  118
        DEFW  0,0,0,0
textline DEFW  0,0,0,0,0,0

        DEFB  0
        DEFB  118
        DEFB  118

space2   EQU   #4301-$
        DEFS   space2

```

```

bl      EQU  blanc*256/256      ; lowbyte of blanc

uw      EQU  upwave*256/256     ; lowbyte upwave

dw      EQU  downwave*256/256   ; lowbyte downwave

displaytab DEFB bl,uw,dw,bl      ; backgroundtable for HR
          DEFB bl,uw,dw,bl

          DEFB bl,uw,dw,bl
          DEFB bl,uw,dw,bl

          DEFB bl,uw,dw,bl
          DEFB bl,uw,dw,bl

          DEFB bl,uw,dw,bl
          DEFB bl,uw,dw,bl

readasci PUSH HL
          SET  7,(HL)             ; invert character
          CALL #7BD
          CP   #1E                ; Newline
          JP   Z,nlpress
          LD   HL,#7D
          ADD  A,L
          LD   L,A
          LD   A,(HL)             ; Translate keypress 2 ascii
          POP  HL
          SET  7,A                ; invert current key
          LD   (HL),A
          INC  HL
          RET

disptime LD  (timecount+1),A      ; set new start
settime  LD  HL,time+4
          INC  (HL)               ; 1/100 sec
settime2 INC  (HL)               ; 2/100 sec
timetest LD  A,(HL)
          CP   38
          JR   NZ,timer
          LD   (HL),28

findtime DEC  HL
          LD   A,(HL)
          CP   28
          JR   C,findtime         ; skip ":"
          JR   settime2

timer    DJNZ settime
          RET

blanc    EQU  $
init     LD   IX,hr              ; Hires mode
          LD   SP,#4400
          LD   H,#3F              ; #3fxx
          LD   D,#BF              ; #bfxx
          LD   E,L
          LDIR                    ; repair 48K bug
          LD   HL,udgdata+1
          LD   DE,#4002
          LD   C,16
copyudg  LD   B,C
          LDI
          INC  HL
          DJNZ copyudg

```



```

LD    B,23
LD    HL,blanc
setblanc LD    (HL),C
INC    HL
DJNZ  setblanc          ; overwrite init as blanc

LD    A,204
LD    L,downwave*256/256
LD    B,23
makewave LD    (HL),A          ; built downwave and upwave
INC    HL
DJNZ  makewave
RRCA
RRCA
JR    NC,makewave-2

LD    DE,udgdata-32      ; also udg screen over init
LD    HL,start
PUSH  HL
LD    HL,udgdata
LD    C,32
JP    #19F9              ; start through ROM

udgdata DEFB 0,56
        DEFB 0,64
        DEFB 0,99
        DEFB 0,80
        DEFB 0,96
        DEFB 0,76
        DEFB 0,67
        DEFB 0,0

        DEFB 0,0
        DEFB 0,200
        DEFB 0,41
        DEFB 0,12
        DEFB 0,10
        DEFB 0,204
        DEFB 0,40
        DEFB 0,7

vars    DEFB 128
?
last    EQU    $

```