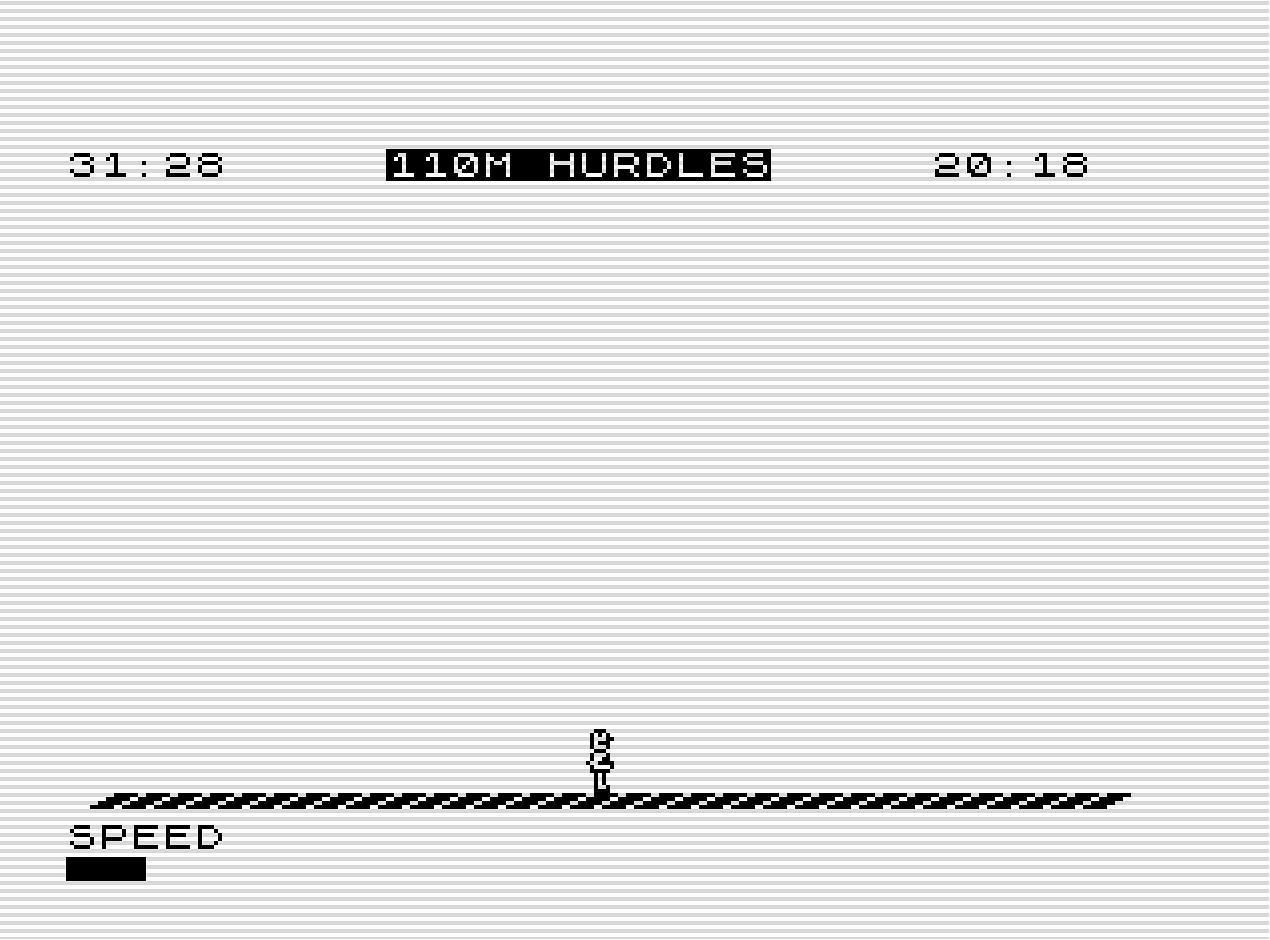
110m Hurdles



Longjump was the base for this game, but in the end a lot of changes where needed to make the game fit 1K. Can you break the worldrecord?

```
; 110m hurdles

? * TORNADO *

           ORG  #4009             ;#4009
           DUMP 49161

lbuf2      EQU  screen+65-16
lbuf3      EQU  lbuf2+26
linedist   EQU  36
nrlines    EQU  138
topspeed   EQU  55

data1      EQU  #4000
fixpos1    EQU  data1+6

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

data2      DEFW 0,0,0             ; the data ABOVE hurdles
fixpos2    DEFW 0,0,0
           DEFW 0,0,0,0,0
           DEFB 0

; speeddata used to show graphical speed
speeddata  DEFB 0,0,0,0,0,0,0,0

           DEFB 0

udgall     DEFB 3,128             ; the head of the man
           DEFB 5,64
           DEFB 4,96
           DEFB 5,192
           DEFB 1

udg1       DEFB 4,76              ; right leg in front
           DEFB 3,148
           DEFB 60,100
           DEFB 64,8
           DEFB 156,112
           DEFB 228,64
           DEFB 12,64
           DEFB 49,32
           DEFB 78,160
           DEFB 80,176
           DEFB 32,136
           DEFB 0,112
           DEFB 1

udg2       DEFB 4,64              ; standing man
           DEFB 3,128
           DEFB 4,64
           DEFB 4,192
           DEFB 9,224
           DEFB 4,32
           DEFB 7,192
           DEFB 2,128
           DEFB 2,128
           DEFB 2,128
           DEFB 2,64
           DEFB 3,192
           DEFB 1

udgjump    DEFB 4,64               ; jumping man
           DEFB 3,128
           DEFB 4,64
           DEFB 4,192
           DEFB 9,224
           DEFB 4,36
           DEFB 127,250
           DEFB 129,2
           DEFB 254,252

hr         LD   BC,#311            ; minimum needed and endmarker
           LD   HL,lowres+#8000    ; the lowres display
           LD   A,#1E
           LD   I,A
           LD   A,#FB
           CALL #2B5               ; show lowres top

           LD   HL,(fixpos1)       ; save original values
           LD   (repair+1),HL      ; for repair later

           LD   B,(HL)             ; timing
           LD   B,6                ; sync hires display
hr00       DJNZ hr00

           LD   B,#90
           LD   D,#40
           LD   C,D                ; on LDI C is never 0
           LD   A,D
           LD   I,A

           LD   HL,screen
nline      LD   A,B                ; get current screenline
           CP   9                  ; test against hurdles
           JR   C,hurdline         ; show data with hurdles
testudg    CP   (HL)               ; test against visible udg
           LD   E,fixpos2*256/256 ; set destination
           JR   NZ,emptyline       ; show NOTHING at all
setudg     INC  L                  ; now point to UDG-data
           LDI                     ; copy udg
           LDI                     ; copy udg
           LD   A,data2*256/256    ; set pointer
           DEC  B                  ; go to next Y
           JP   lbuf+#8000         ; show current line

hurdline   SUB  (HL)               ; test against UDG and hurdles
           LD   E,fixpos1*256/256
           JR   NZ,hurdonly        ; show only hurdles
           INC  HL                 ; point to UDG-data
           LDI                     ; copy udg
           LDI                     ; copy udg
db         DEC  B                  ; go to next Y
           JP   lbuf+#8000         ; show current line

hurdonly   LD   A,(HL)             ; filler
```

```
           XOR  A                  ; set hurdlepointer
           RET  NZ                 ; filler
           RET  NZ                 ; filler
           JR   db                 ; filler and exit through

emptyline  PUSH HL                 ; just do 207 tstates
           POP  HL                 ; of nothing
           LD   A,0
           LD   (DE),A             ; clear possible old udg
           INC  DE
           LD   (DE),A
           DEC  B                  ; go to next Y

; init cleared after start
lbuf       LD   R,A
init       LD   IX,hr              ; 04 Hires mode
           LD   SP,#4400           ; 07

           LD   H,#3F              ; 09 #3fxx
           LD   D,#BF              ; 11 #bfxx
           LD   E,L                ; 12
           LDIR                    ; 14 repair 48K bug
           LD   HL,lbuf+2          ; 17
           LD   (HL),C             ; 18
           JP   initcont           ; 21
           DEFB 0,0                ; 23
           JP   NZ,nline           ; 48K bug
           JP   exit               ; 48K bug

exit       LD   A,lbuf2/256        ; buffer is also data
           LD   I,A
           RET  NZ                 ; timing

           LD   B,4
showpath   LD   A,lbuf2*256/256+2 ; show ground, 4x
           CALL lbuf2+#8000        ; but 1 t state shifted
           PUSH HL
           POP  HL
           DEC  B
           JP   NZ,showpath

lr2        LD   HL,lowres2+#8000  ; show SPEEDtext
           LD   BC,#203
           LD   A,#1E
           LD   I,A
           LD   A,#E9
           CALL #2B5

           LD   C,6                ; preload and delay
           EX   (SP),HL
           EX   (SP),HL

           LD   A,#40
           LD   I,A                ; timing
           LD   I,A
           LD   A,speeddata*256/256-1

speeddisp  LD   B,8
lr3        DJNZ lr3

           INC  L                  ; timing
           DEC  HL
           CP   (HL)
           LD   R,A
           CALL lbuf3+#8000       ; show graphical speed
           DEC  C                 ; show it 6x
           JR   NZ,speeddisp

           LD   HL,fixpos1
fptest     LD   A,0
           OR   A
           JR   Z,repair          ; no hurdle, erase man
           CP   (HL)
           JR   Z,repair          ; jumped

colfound   LD   (collided+1),A    ; set collission signal

repair     LD   HL,0
           LD   (fixpos1),HL

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

disptime   LD   HL,frames         ; point to time sysvar
timecount  LD   A,0               ; what was previous
           SUB  (HL)
           RET  Z                 ; no change in time
           LD   B,A
           LD   A,(HL)
           LD   (timecount+1),A   ; save new start
settime    LD   HL,time+4
           INC  (HL)              ; 1/100 time added
settime2   INC  (HL)              ; 1/50
timetest   LD   A,(HL)
           CP   38
           JR   Z,findtime
           DJNZ settime
           RET

findtime   LD   (HL),28
           DEC  HL
           LD   A,(HL)
           CP   28
           JR   C,findtime+2      ; test ":"
           CP   118
           JR   NZ,settime2       ; 100 sec times passed
           POP  HL                ; drop return

finish     LD   A,118
           LD   (startmsg),A
           LD   A,%10111111
           IN   A,(254)
           RRA
           JR   C,finish

gameloop   LD   A,10              ; set all hurdles
           LD   (nrhurd+1),A
           LD   HL,800            ; set distance to run
           LD   (nextmove+1),HL
           XOR  A
           LD   (nextudg+1),A
           LD   (setspeed+1),A

           LD   HL,#1C1C          ; reset previous time
           LD   (time),HL
           LD   (time+3),HL

           LD   HL,#4018
clrhurd    LD   (HL),A
           DEC  L
           JR   NZ,clrhurd
           LD   L,#10
           INC  (HL)              ; set first hurdle in sight

           CALL makescreen

           LD   HL,onyrmarks      ; READY and GO text
starter    LD   A,(frames)
           SUB  100
           LD   E,A

wst        CALL readkey           ; both keyboard and zxpand
           JR   NZ,finish         ; false start

           LD   A,(frames)
           CP   E
           JR   NZ,wst            ; test 2 sec delay

           LD   DE,startmsg       ; show on screen
jrldi      LD   A,(HL)            ; write text
           LDI
           CP   118
           JR   NZ,jrldi          ; until NL found
           LD   A,(HL)
           INC  A
           JR   NZ,starter        ; do all texts

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

playloop   CALL disptime          ; show time passed

jumpcnt    LD   A,1
           LD   HL,nextudg+1
           DEC  A
           JR   NZ,jumpfase       ; in air
           CALL readkey           ; read keys
           LD   HL,setspeed+1
           JR   Z,nomove          ; no key pressed
prevmove   CP   0
           JR   Z,nomove          ; new key must be different
           LD   (prevmove+1),A
           BIT  0,A
           JR   NZ,jumpkey        ; do a jump
domove     LD   A,(HL)
           CP   topspeed
           JR   NC,onground
           INC  (HL)              ; add speed
           INC  (HL)
nomove     LD   A,(HL)
           OR   A
           JR   Z,onground        ; not below 0
           DEC  (HL)
onground   LD   A,16              ; feet on ground
           LD   HL,nextudg+1
           RES  7,(HL)            ; show run udg
           JR   keydone

jumpkey    LD   HL,nextudg+1
```

```
           SET  7,(HL)            ; show jump udg
           LD   A,3               ; time in air
jumpfase   LD   (jumpcnt+1),A
           LD   A,21              ; y-pos in air
keydone    LD   (xy+1),A

           CALL setspeed          ; speedindicator
           CALL makescreen

           LD   (collided+1),A    ; signal no collission, A=0

           LD   A,(setspeed+1)
           AND  #F8
           RRCA
           RRCA
           LD   B,A

           JR   Z,linemoved

           LD   A,(nextudg+1)     ; swap running udg
           XOR  1
           LD   (nextudg+1),A

nextmove   LD   HL,0              ; do a step
           DEC  HL
           LD   (nextmove+1),HL
           LD   A,H
           OR   L
           JR   Z,finline         ; finish reached
           LD   HL,#4018          ; point to hurdledata
shift      RL   (HL)              ; let hurdle come in
           PUSH AF
           JR   NC,nofield        ; pass hurdle to next field
           LD   A,L
           SUB  16                ; is it on field 16
           JR   NZ,nofield
nrhurd     LD   A,10              ; if so set new hurdle
           DEC  A                 ; if still remaining
           JR   Z,nofield
sethurd    LD   (nrhurd+1),A
           LD   A,1
           LD   (#4018),A         ; set next hurdle
nofield    LD   A,fixpos1*256/256
           CP   L
           JR   NZ,nfield
           LD   A,(HL)            ; fetch field under player
           LD   (fptest+1),A
nfield     POP  AF
           DEC  L
           JP   P,shift           ; do full line shifting
           DJNZ nextmove

linemoved  LD   HL,frames         ; the usual delay to keep
           LD   A,(HL)            ; game playable
           SUB  4
wfr        CP   (HL)
           JR   NZ,wfr

collided   LD   A,0               ; test hurdle hit
           OR   A                 ; this is set during HRdisplay
           JR   Z,pllp

           LD   HL,0              ; erase hurdle
           LD   (fixpos1),HL
```

```
            LD   A,(setspeed+1)    ; get current speed
            LD   L,A
            RRA                    ; speed = 0.5 speed
            ADD  A,L               ; speed = 1.5 speed
            RRA                    ; speed = 0.75 speed
            LD   (setspeed+1),A    ; save reduced speed

pllp        JP   playloop

finline     CALL disptime          ; add final time passed
            LD   HL,time-1         ; and do worldrec test
            LD   DE,worldrec-1
            LD   BC,6
same        INC  HL
            INC  DE
            DEC  C
            JR   Z,finjp
            LD   A,(DE)
            CP   (HL)
            JR   Z,same
            JR   C,finjp
            LDIR
finjp       JP   finish

readkey     XOR  A
            LD   D,A
            IN   A,(254)           ; read keyboard
            CPL
            AND  31
            JR   Z,zxpand          ; test zxpand when no key

            LD   A,%11100111       ; keyboard jump is numeric
            IN   A,(254)
            CPL
            AND  31
            JR   Z,nofire
            INC  D
nofire      LD   A,%01011000       ; skip NL + numeric for speed
            IN   A,(254)
            CPL
            AND  30
            ADD  A,D
            RET

zxpand      LD   BC,%1110000000000111
            LD   A,#A0
            OUT  (C),A
            JR   zx2
zx2         IN   A,(C)
            RRA
            RRA
            RRA
            CPL
            AND  31
            RET

setspeed    LD   A,0               ; fetch speednumber
            LD   HL,speeddata-1    ; make it graphical
fillfull    INC  HL
            SUB  8
            LD   (HL),255          ; each 8 filled
            JR   NC,fillfull
```

```
fullfill   ADC  A,8
           LD   (HL),0          ; clear next field
ff         SCF                  ; rotate in carry
           RR   (HL)
           DEC  A
           JR   NZ,ff
           DEFB 254             ; CP n
filler     LD   (HL),A
           INC  HL
           CP   (HL)            ; clear remaining speed
           JR   NZ,filler
           RET

makescreen PUSH DE
           LD   DE,screen
xy         LD   C,#10           ; Y pos of man
           LD   HL,udgall       ; head
           CALL makeudg
nextudg    LD   A,3             ; find correct UDG to display
           ADD  A,A
           LD   L,udgjump*256/256
           JR   C,udgok
           BIT  1,A
           LD   L,udg2*256/256
           JR   Z,udgok
           LD   L,udg1*256/256
udgok      CALL makeudg
           XOR  A
           LD   (DE),A          ; set impossible line
           POP  DE
           RET

makeudg    LD   A,C
           LD   (DE),A
           INC  DE
           INC  BC
           LDI
           LDI
           LD   B,(HL)
           DJNZ makeudg
           RET

n          EQU  27
x          EQU  101

lowres2    DEFB 118
           DEFB "S"-n,"P"-n,"E"-n,"E"-n,"D"-n

lowres     DEFB 118

time       DEFB 28,28,14,28,28
           DEFW 0,0
           DEFB 0,157,157,156,"M"+x
           DEFB 128,"H"+x,"U"+x,"R"+x,"D"+x
           DEFB "L"+x,"E"+x,"S"+x,0
           DEFW 0,0
worldrec   DEFB 30,28,14,29,36
           DEFB 118
           DEFW 0,0,0,0,0,0
           DEFB 0
startmsg   DEFW 0,0,0
           DEFB 118

onyrmarks  DEFB "R"-n,"E"-n,"A"-n,"D"-n,"Y"-n
```

```
           DEFB 118,0,0
           DEFB "G"-n,"O"-n
           DEFB 118,255

space      EQU  #4400-110-$

           DEFS space

screen     DEFB 0                 ; the screenmemory

initcont   LD   D,H               ; clear LBUF
           LD   E,L
           INC  DE
           LD   C,22
           LDIR

           LD   HL,#4FED          ; "LD R,A"
           LD   (lbuf2),HL
           LD   HL,lbuf2+2
           LD   B,32
setlb      LD   (HL),63           ; 32 displaydata and buffer
           INC  HL
           DJNZ setlb
           LD   (HL),201          ; "RET"

           LD   HL,#4018          ; clear man memory
clsini     DEC  L
           LD   (HL),C
           JR   NZ,clsini

           JP   gameloop

vars       DEFB 128
; after screen LBUF2 for ground is made
; end of memory is stack

?
last       EQU  $
```