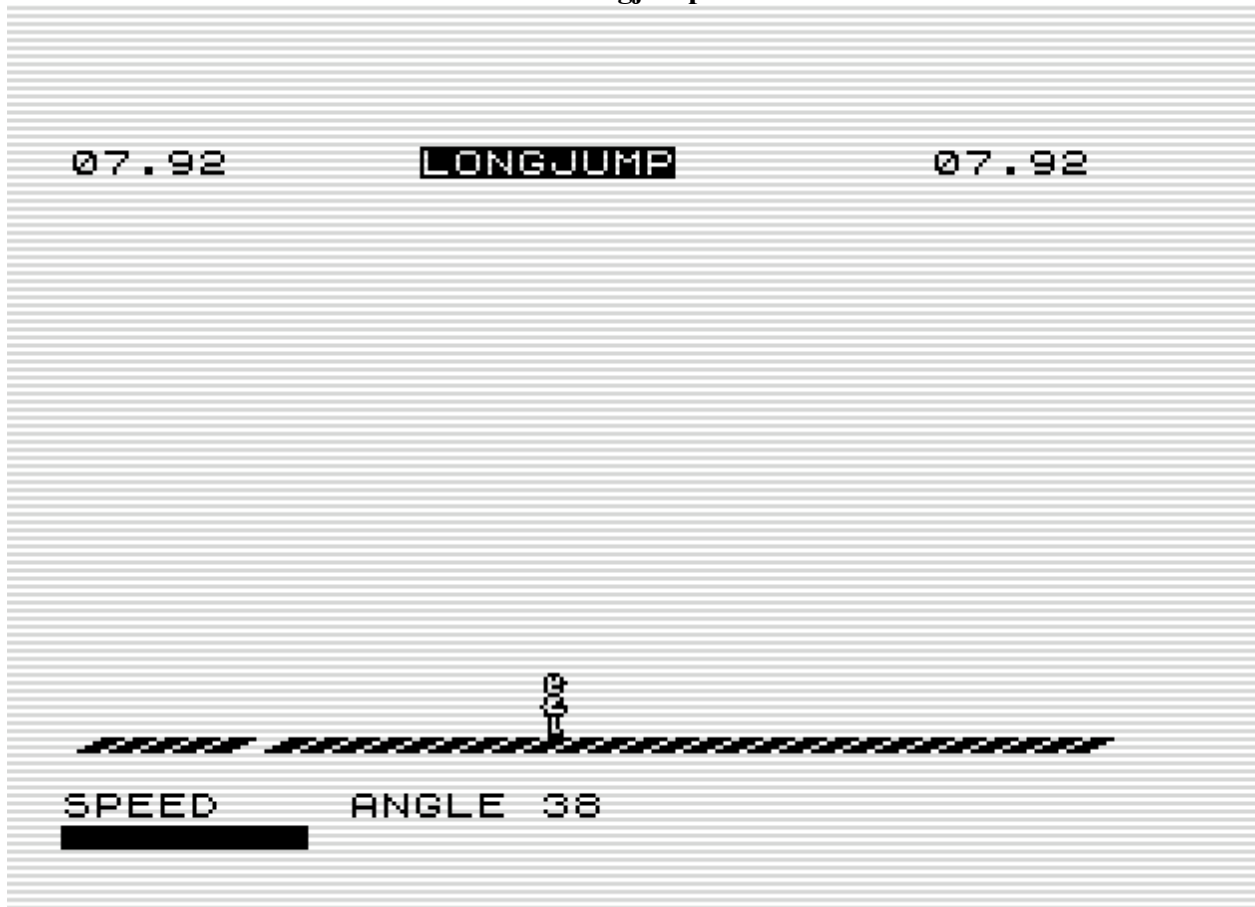


## Longjump



**I found a way to display at least 2 characters on a line and erase them for the next. With some nice animation it is possible to code this longjump. A new formula made it possible to add ZXPAND-joystick support.**

```
; Longjump

? * TORNADO *

                ORG   #4009                ;#4009
                DUMP  49161

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

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

; line reused to make the bottomline
line      LD    HL,#4FED        ; "LD R,A"
         LD    (lbuf2),HL
         LD    HL,lbuf2+2
         LD    B,32
setlb     LD    (HL),C          ; 32 spaces
         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

tab        DEFB 252,243,207,63,0 ; line filler

; speeddate reused to show graphical speed
speeddata DEFB 254,1,248,7,224,31,128,127

         DEFB 0

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

udg1       DEFB 4,76            ; right leg in front
         DEFB 3,148
         DEFB 60,100
         DEFB 64,72
         DEFB 158,112
         DEFB 228,64
         DEFB 14,64

```

```

        DEFB 51,32
        DEFB 78,160
        DEFB 80,176
        DEFB 32,136
        DEFB 0,112

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

udg3    DEFB 4,76          ; left leg in front
        DEFB 3,148
        DEFB 60,100
        DEFB 68,8
        DEFB 156,240
        DEFB 228,64
        DEFB 12,192
        DEFB 49,160
        DEFB 78,160
        DEFB 80,176
        DEFB 32,136
        DEFB 0,112

udgjump DEFB 4,64,3,128    ; landing
        DEFB 4,64,4,192
        DEFB 9,224,4,36
        DEFB 7,202,2,58
        DEFB 3,2,0,252
        DEFB 0,0,0,0

hr      LD  HL,lowres+#8000 ; the lowres display
        LD  BC,#211        ; minimum needed
        LD  A,#1E
        LD  I,A
        LD  A,#FB
        CALL #2B5

hr00    LD  B,7            ; sync hires display
        LD  E,B            ; preset for any clear
        DJNZ hr00

        LD  D,#40
        LD  A,D
        LD  I,A
        XOR A              ; reset value

        LD  HL,screen
        LD  C,nrlines      ; most empty lines

lowret  LD  (DE),A          ; erase displayed udg
        DEC E
        LD  (DE),A

nline   LD  A,(HL)          ; udgline reached
        SUB C              ; test so and set A=0

```

```

        JP    NZ,empty          ; or show empty line
        NOP                      ; fixed x-pos now
        LD    E,3               ; fetch x-pos
        INC   L
        LDI                      ; copy udg
        LDI                      ; copy udg
        INC   C                 ; C 1 less, not 2
        JP    lbuf+#8000        ; show line

lbuf    LD    R,A
; init cleared after start
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,init
        LD    (HL),C
        JP    initcont
        DEFB 0
        JP    NZ,lowret         ; 48K bug
        JP    exit              ; 48K bug

empty   LD    B,4
loop    PUSH  HL
        LD    A,(HL)
        POP  HL
        DJNZ loop              ; do 207 Tstates nothing
        LD    E,C
        DEC  C
        JR   NZ,nline
        RET  NZ                 ; timing in sync
        JP   exit3

exit    LD    (DE),A            ; clear final UDG displayed
        DEC  E
        LD    (DE),A

exit3   LD    B,4
showpath LD A,line*256/256      ; show ground, 4x
        CALL lbuf2+#8000        ; but 1 tstate shifted
        PUSH HL
        POP  HL
        DEC  B
        JP   NZ,showpath

1r2     LD    HL,lowres2+#8000  ; show text
        LD    BC,#209
        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

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

restart   LD A,2                    ; reset speed previous jump
        LD (stepspeed+1),A        ; do a step to clear ground
        LD HL,#1C1C
        LD (angle),HL             ; reset angle on screen
        LD (text+3),HL           ; reset cm
        LD (text),HL             ; reset m
        LD A,linedist             ; set jumpline out of screen
        LD (jumpline+1),A
        XOR A
        LD (angledec+1),A         ; reset angle in memory
        LD (jumpsteps+1),A        ; reset jumped steps

gameloop CALL makescreen          ; make a screen

        LD A,(jumpline+1)
        CP 13
        JP Z,wk                  ; not jumped in time, false try

; read keys, add speed
        CALL readinput            ; single point of play input

        BIT 0,D
        JR NZ,jumppress          ; start jumping

        LD HL,stepspeed+1

        LD A,D
        OR A
        JR Z,nomove
prevmove CP 0
        LD (prevmove+1),A         ; set new key pressed
        JR Z,nomove              ; new must be different
domove   LD A,(HL)
        CP topspeed
        JR NC,wait                ; topspeed reached
        INC (HL)                  ; add speed
        INC (HL)

nomove   LD A,(HL)
        OR A
        JR Z,wait                ; do not go below 0
        DEC (HL)                  ; slow down power

wait     LD A,(HL)                ; fetch speednumber
        LD HL,speeddata           ; make it graphical
fillfull SUB 8
        JP C,fullfill            ; only remainder
        LD (HL),255              ; each 8 filled
        INC HL

```

```

        JR    fillfull

jumppress LD    A,3                ; set jumping UDG
        CALL makescreen-3

jump2    LD    B,1                ; delay needed
        CALL waitloop

        CALL readinput            ; again read input
        BIT    0,D
        JR    Z,jumpfase         ; angle set, do actual jump

        LD    HL,angledec+1       ; angle decimal notation
        INC    (HL)               ; add 1 to decimal notation

        LD    HL,angle+1

addangle INC    (HL)              ; add 1 to visible notation
        LD    A,(HL)
        CP    38
        JR    NZ,jump2
        LD    (HL),28
        DEC    HL
        INC    (HL)
        CALL makeline
        LD    HL,xy+1             ; each 10 go 1 up on screen
        INC    (HL)
        JR    jump2

jumpfase LD    D,2
        DEC    D                  ; up or down counter
move     CALL makescreen          ; built a screen
        PUSH DE
        CALL makeline            ; move the bottom
        POP  DE
        LD    B,6
        CALL waitloop            ; delay
        LD    HL,xy+1
        LD    A,(HL)
        ADD    A,D                ; go up or go down with player
        LD    (HL),A
        SUB    17
        SUB    18
        JR    C,move              ; test top or bottom reached
        SUB    D                  ; below zero and odd
        DEC    D
        LD    (nextudg+1),A       ; set landing UDG
        JR    Z,move-1           ; on zero move down

        CALL makescreen          ; set landed man

; calculate distance and worldrec?
angledec LD    A,0
        CP    45
        LD    E,A
        JR    C,mul0             ; angle <= 45 degrees
        JR    NZ,min90
        LD    HL,stepspeed+1
        INC    (HL)               ; on perfect angle add 2 steps
        INC    (HL)               ; for perfect worldrec
min90    LD    A,90
        SUB    E                  ; read 90-angle
        JR    C,wk               ; angle above 90 deg is false try
mul0     LD    HL,0

```

```

LD    D,H
LD    E,A
LD    A,(stepspeed+1)    ; always more than 1
INC    A
DEFB 254                ; skip first addition
mul    ADD    HL,DE
DEC    A
JR     NZ,mul            ; result = angle * speed

jumpsteps LD    B,0                ; now multiply by steps
INC    B                ; behind line
LD    A,H
ADD    A,A
ADD    A,A                ; but divide by 64 too
LD    E,A
LD    HL,0
LD    D,H
DEFB 254                ; again skip first addition
stepmul ADD    HL,DE
DJNZ  stepmul
EX     DE,HL

de0     LD    A,D
OR      E
JR     Z,wrtest        ; distance calculated, landed before line
LD    HL,text+5
pause   DJNZ  pause        ; nice measurement effect
DEFB #3A                ; hide ten
ten     LD    (HL),28
dot     DEC    HL
LD    A,(HL)
CP     27
JR     Z,dot            ; skip dot when found
INC    (HL)
CP     37
JR     Z,ten
DEC    DE
JR     de0

wrtest  LD    HL,text-1        ; current jump
LD    DE,worldrec-1        ; current worldrecord
LD    C,6
same    INC    HL
INC    DE
DEC    C
JR     Z,wk
LD    A,(DE)
CP     (HL)
JR     Z,same            ; still the same
JR     NC,wk            ; smaller
LDIR                     ; new record

wk      LD    A,%10111111    ; read G-NL
IN     A,(254)
CPL
AND    31
JR     Z,wk            ; restart keyboard only
JP     restart        ; next game

readinput XOR    A                ; read full keyboard
LD    D,A                ; signal no keys pressed
IN     A,(254)
CPL
AND    31

```

```

        JR    Z,zxpend            ; nothing pressed, test ZXPAND

        LD    A,%01000110        ; all but Q-T, A-G, H-NewL
        IN    A,(254)            ; test jump key pressed
        CPL
        AND    31
        JR    Z,nofire
        INC    D                ; bit 0 used for fire

nofire   LD    A,%11111001        ; Q-T and A-G
        IN    A,(254)            ; test keyboard
        CPL
        AND    30                ; just accept WSEDRFTG
        ADD    A,D                ; otherwise fire gets lost
        LD    D,A
        RET

zxpend   LD    BC,%1110000000000111
        LD    A,#A0
        OUT    (C),A            ; activate joystick ZXPAND
        JR    zx2                ; delay
zx2      IN    A,(C)              ; read joystick
        RRA                      ; shift 3 bits
        RRA
        RRA
        CPL
        AND    31
        LD    D,A                ; keys to D, bit 0 - fire
        RET                      ; other bits are directions

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, skip clearing
filler   LD    (HL),A
        INC    HL
        CP    (HL)                ; clear remaining speed
        JR    NZ,filler

        LD    B,8
        CALL waitloop

stepspeed LD    A,2
        OR    A
        JR    Z,still            ; no speed, reset startudg

        CALL makeline

; when speed alter udgpointer to show animation
        LD    A,(nextudg+1)        ; get current udg
        INC    A                ; point to next
        AND    3                ; 0-3
still    LD    (nextudg+1),A        ; set next udg
        JP    gameloop            ; continue game

makeline LD    DE,line
        LD    H,D

metpoint LD    A,0                ; set next bit pattern
        INC    A
        AND    3
        LD    (metpoint+1),A

```



```

        ADD  A,taddr*256/256
        LD   L,A                ; point to correct linebyte

nolineset LD   A,(HL)            ; get linebyte
        LD   B,32
setline  LD   (DE),A
        INC  DE                ; show full line
        DJNZ setline

        LD   A,L
        LD   HL,jumpline+1
        SUB  taddr*256/256
        JR   NZ,nodec
        DEC  (HL)              ; move jumpline

nodec    LD   E,(HL)
        LD   H,D
        ADD  A,A
        ADD  A,24
        LD   L,A

jumpline LD   A,linedist
        CP   13
        JR   NC,notjumped      ; only possible when jumping

        PUSH HL
        LD   HL,jumpsteps+1    ; count steps during jump
        INC  (HL)
        POP  HL

notjumped CP  31
        RET  NC                ; jumpline not in sight

        LD   A,line*256/256
        ADD  A,E
        LD   E,A              ; point to table value
        LDI                ; show jumpline
        LDI
        RET

waitloop LD   HL,frames        ; the ZX81 is still too fast
        LD   A,(HL)            ; so this game has delay too
        SUB  B
wfr      CP   (HL)
        JR   NZ,wfr
        RET

        LD   (nextudg+1),A
makescreen PUSH DE
        LD   DE,screen
xy        LD   BC,#0410        ; Y pos of man
        LD   HL,udgall        ; head

setudg    LD   A,C              ; y-pos on screen
        LD   (DE),A
        INC  DE
        INC  C                ; undo DEC BC from LDI
        LDI
        LDI                  ; but C must be 1 less
        DJNZ setudg

nextudg   LD   A,0              ; find correct UDG to display
        BIT  0,A

```

```

step2      LD    L,udg2*256/256      ; standing man on 0 and 2
           JR    Z,udgok
           LD    L,udg1*256/256      ; left leg udg
           DEC   A
           JR    Z,udgok
           ADD   A,A
           LD    L,udg3*256/256      ; landing udg
           JR    NC,udgok
           LD    L,udgjump*256/256   ; right leg udg

udgok      LD    B,12                ; the final 12 lines
           LD    A,screen*256/256+64-16
           SUB   E
           JR    NZ,setudg           ; test end of screen reached
           LD    (DE),A              ; set impossible line
           POP   DE
           RET

n          EQU   27
x          EQU   101

lowres2    DEFB  118
           DEFB  "S"-n,"P"-n,"E"-n,"E"-n,"D"-n,0,0,0,0
           DEFB  "A"-n,"N"-n,"G"-n,"L"-n,"E"-n,0
angle      DEFB  28,28

lowres     DEFB  118

text       DEFB  28,28,27,28,28
           DEFW  0,0,0
           DEFB  0,"L"+x,"O"+x,"N"+x,"G"+x
           DEFB  "J"+x,"U"+x,"M"+x,"P"+x,0
           DEFW  0,0,0
worldrec   DEFB  28,28,27,28,28
           DEFB  118

space      EQU   #4400-108-$

           DEFS  space
? ERROR Label?

screen     EQU   $                  ; the screenmemory
initcont   LD    D,H                  ; clear LBUF
           LD    E,L
           INC   DE
           LD    C,21
           LDIR
           LD    HL,tab                ; reuse part of sysvar
           LD    DE,taddr
           LD    C,4
           LDIR
           LD    E,24                  ; linetable over sysvar
           LD    HL,speeddata
           LD    C,8
           LDIR
           JP    line

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

?
last       EQU   $

```