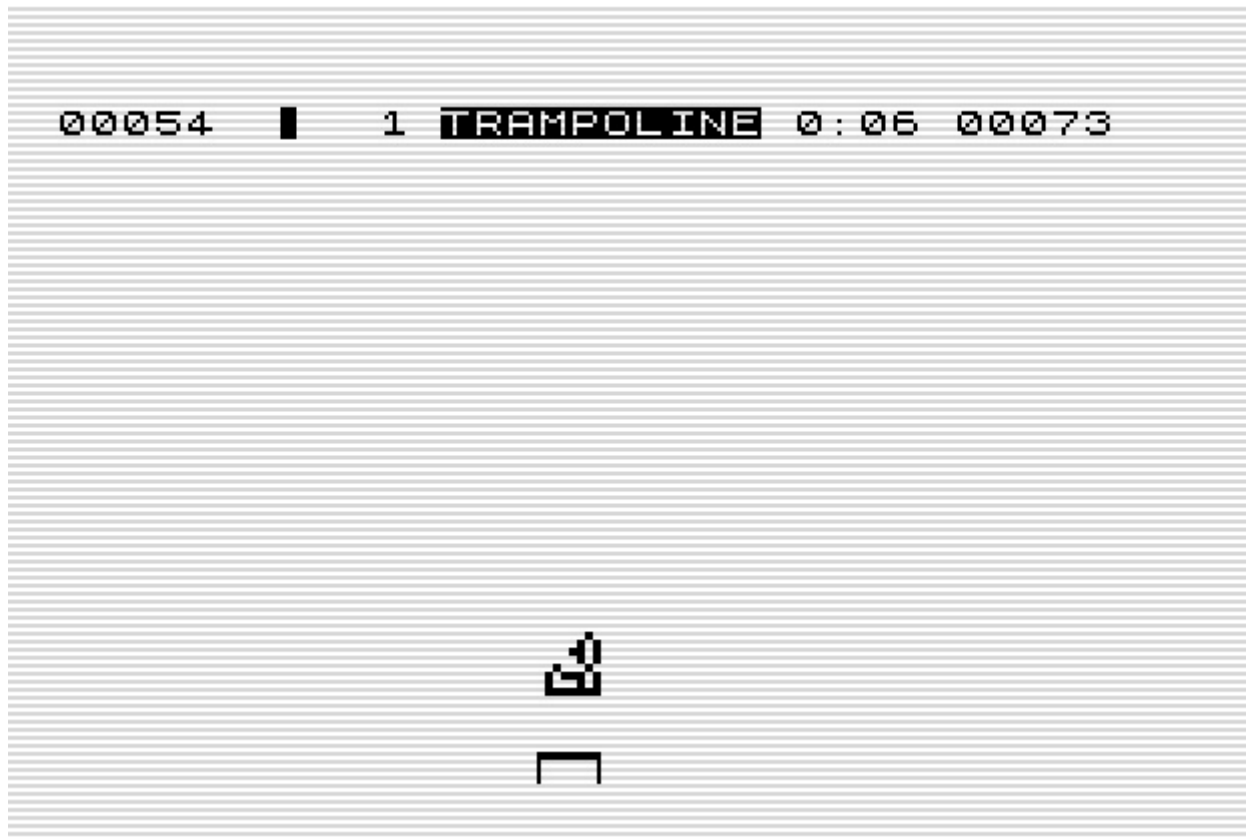


## Trampoline



**I wanted a game with smooth scrolling. It has been too long that I used it. Most games had bitwise movement. Then it hit me... a trampoline game. I searched on games but couldn't find it on 8 bit machines. So this was a good game to code.**

```
;; Trampoline
; Game 73 in 1K hires for the ZX81.

? * TORNADO *

                ORG   #4009                ;#4009
                DUMP  49161

; program starts here, both BASIC and machinecode
basic          EX    AF,AF'                ; delay intrupt,opcode no bit6
                LD    H,B                  ; preset for 48K bug to #40
                JR    init0                ; continue where room

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

                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

                DEFB  0,0,0
```

```

; 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      LD    IX,hr           ; hr lowbyte bit 5 reset
; lowbyte over flagx which resets bit 5 on load
; HR must be set on right address or game crashes

movesdone  LD    E,L             ; DE now #xx.L

taddr      DEFW 0                ; used by ZX81 on LOAD only
; unharmed code

          LD    B,4              ; copy >1K code

frames     DEFB #16+1            ; LD D,n , after LOAD -1
          DEFB #C0                ; highbyte must have bit 7 set
coords     LDIR                  ; DE now #C0.L = H1 + #8000
; fix 48K bug before display
prcc       JP    init            ; continue to mainprog

cdflag     DEFB 64                ; used by ZX81

; Place ANY code to fill up to #4040

eline      LD    A,E
          LD    E,4
empty      DEC    E
          JR    NZ,empty
          INC    DE
          JR    emptyin

lbuf       LD    R,A
          DEFW 0
          RET    NC

; some lowres, HR must start AFTER #403F, but before #4070
hr         LD    HL,lowres+#8000  ; the lowres display
          LD    BC,#211           ; minimum needed #11
          LD    A,#1E
          LD    I,A
          LD    A,#FB
          CALL  #2B5              ; show lowres screen

          LD    B,9
hr00       DJNZ hr00

; the hr part
          LD    BC,#A0FF

          LD    HL,screen

line       LD    A,(HL)
          XOR    B
          LD    DE,nxtlin
          JR    NZ,eline
          INC    HL
          LD    A,D
          LD    I,A
          LD    A,E
          LDI

```

```

        LDI
        EX    (SP),HL
        EX    (SP),HL
emptyin  CALL  lbuf+#8000
        PUSH HL
        POP  HL
        OR   (HL)
        DJNZ line

; fixed end of HR-routine
        CALL #292           ; back from intrupt
        CALL #220
        LD   IX,hr
        JP   #2A4

hitab    DEFB 18,21,24,28,32,37,43,50,58,68,80

mleft    DEFB 0,192
        DEFB 3,48
        DEFB 15,48
        DEFB 3,48
        DEFB 0,192
        DEFB 3,240
        DEFB 3,48
        DEFB 3,48
        DEFB 12,48
        DEFB 3,240
        DEFB 3,48
        DEFB 3,48
        DEFB 12,48
        DEFB 3,240
        DEFB 0,0

keytab    DEFB 10-1           ; Q
        DEFB fwsal*256/256
        DEFB #83
        DEFB 5-1             ; A
        DEFB scrleft*256/256
        DEFB #22
        DEFB 31-1            ; L
        DEFB scrright*256/256
        DEFB #12
        DEFB 25-1            ; P
        DEFB bwsal*256/256
        DEFB #45

falsek    DEFB 0

fwsal     DEFW fw1
        DEFW fw2
        DEFW fw3
        DEFW fw4
        DEFW mleft

scrleft    DEFW mfront
        DEFW mright
        DEFW mback
        DEFW mleft

scrright   DEFW mback
        DEFW mright
        DEFW mfront
        DEFW mleft

```

bwsal	DEFW	fw4
	DEFW	fw4
	DEFW	fw3
	DEFW	fw3
	DEFW	fw2
	DEFW	fw2
	DEFW	fw1
	DEFW	mleft
tramp	DEFB	255,255
	DEFB	128,1
fw1	DEFB	63,63
	DEFB	192,195
	DEFB	63,63
	DEFB	12,51
	DEFB	0,51
	DEFB	0,195
	DEFB	0,63
	DEFB	0,0
fw2	DEFB	255,252
	DEFB	204,12
	DEFB	207,204
	DEFB	48,48
	DEFB	204,0
	DEFB	207,0
	DEFB	204,0
	DEFB	48,0
fw3	DEFB	252,0
	DEFB	195,0
	DEFB	204,0
	DEFB	204,48
	DEFB	252,252
	DEFB	195,3
	DEFB	252,252
	DEFB	0
fw4	DEFB	0,12
	DEFB	0,51
	DEFB	0,243
	DEFB	0,51
	DEFB	12,12
	DEFB	51,243
	DEFB	48,51
	DEFB	63,255
mfront	DEFB	7,224
	DEFB	24,24
	DEFB	26,88
	DEFB	24,24
	DEFB	7,224
	DEFB	31,248
	DEFB	24,24
	DEFB	24,24
	DEFB	97,134
	DEFB	31,248
	DEFB	25,152
	DEFB	25,152
DEFB	97,134	

```

        DEFB 126,126
        DEFB 0

mright  DEFB 0,48
        DEFB 0,204
        DEFB 0,207
        DEFB 0,204
        DEFB 0,48
        DEFB 0,252
        DEFB 0,204
        DEFB 0,204
        DEFB 0,195
        DEFB 0,252
        DEFB 0,204
        DEFB 0,204
        DEFB 0,195
        DEFB 0,252
        DEFB 0,0

mback   DEFB 7,224
        DEFB 31,248
        DEFB 31,248
        DEFB 31,248
        DEFB 7,224
        DEFB 31,248
        DEFB 24,24
        DEFB 24,24
        DEFB 97,134
        DEFB 31,248
        DEFB 25,152
        DEFB 25,152
        DEFB 97,134
        DEFB 126,126
        DEFB 0,0

eog      LD  HL,score-1      ; your score
        LD  DE,hiscore-1    ; hiscore
        LD  BC,6             ; size 5

fihi     INC  HL
        INC  DE
        DEC  C               ; when C=0
        LD  A,(DE)           ; (de) = #76
        CP  (HL)             ; (hl) = 0
        JR  Z,fihi           ; so not same
        CALL C,#19F9         ; and no hiscore

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

        LD  SP,#4400
        LD  (frames),A
        LD  (allbits+1),A

        LD  HL,score
        LD  C,28
        LD  (HL),C
        INC  HL
        CP  (HL)
        JR  NZ,clsc

clsc      LD  B,4
        INC  HL

```

```

LD    (HL),A
DJNZ  clbit

LD    L,time*256/256
LD    (HL),C
INC   HL
INC   HL
cltime LD    (HL),C
INC   HL
CP    (HL)
JR    NZ,cltime

LD    HL,mleft
LD    (what+1),HL
INC   A
LD    (nrmoves+1),A    ; set nr moves to 1
LD    A,29
LD    (mulfact),A

LD    BC,#1201          ; start on trampoline, move up
LD    DE,hitab

playloop LD    A,(time)
CP     30
JR     Z,eog            ; time passed

LD    A,(DE)
CP     18               ; on trampoline
JR     Z,readkey
CP     B
JR     NZ,updown        ; top reached?
LD     C,255            ; move down

updown  LD    A,B
ADD    A,C
LD     B,A              ; move up or down

; a move is allowed
readkey  EXX
LD     BC,(lastk)
LD     A,C
INC    A

CALL   NZ,#7BD
EXX

DEC    A
JP     Z,dokey          ; JUMP needs no test

LD     HL,(what+1)      ; is a move on going
PUSH   DE
LD     DE,mleft         ; test standing man
OR     A
SBC    HL,DE
POP    DE
LD     L,A
JR     NZ,nexttst       ; next step of current move

old     LD    A,0
OR     A
JR     Z,nokey
SUB    L                ; sub 1  make it not old

JP     NZ,dokey         ; no other move allowed

```

```

nokey      LD      A,L
           LD      (falsek),A          ; store current key as false
           LD      HL,keytab-1
fkey       INC     HL
           CP      (HL)
           INC     HL
           INC     HL
           JR      NZ,fkey
           LD      A,(HL)              ; get score and bit tester
           AND     7                   ; score only
           LD      (scp+1),A           ; set possible score
           LD      A,(HL)
           AND     #F0                 ; get bit tester
           LD      (bitset+1),A        ; set tester
           DEC     HL
           DEC     HL
           LD      A,L
           CP      falsek*256/256      ; is it a false key?
           LD      A,(HL)              ; get key pressed
           JR      Z,dokey             ; invalid keypressed
           LD      (old+1),A           ; store as only move allowed
           INC     HL
           LD      L,(HL)              ; get table start
           JR      keyin

nexttst    LD      A,0
           XOR     D
           LD      (nexttst+1),A
           JR      NZ,dokey
           LD      A,L
nextstep   LD      HL,0
           INC     HL
keyin      PUSH    DE
           LD      D,(HL)
           INC     HL
           LD      (nextstep+1),HL
           LD      H,(HL)
           LD      L,D
           LD      (what+1),HL
           LD      DE,mleft
           OR      A
           SBC     HL,DE
           JR      NZ,dokey-1
           LD      HL,movesdone
           INC     (HL)

           PUSH    AF
nrmoves    LD      A,(HL)              ; movesdone
           CP      0
           LD      HL,mulfact
           JR      C,scadd             ; not reached

bitset     LD      A,0
allbits    OR      0
           LD      (allbits+1),A
           EXX
           PUSH    AF
           LD      B,4
           LD      HL,compmove
setbits    RLCA
           LD      (HL),0
           JR      NC,nextbit
           LD      (HL),5
nextbit    INC     HL

```

```

        DJNZ setbits
        POP AF
        EXX
        SUB #F0
        JR NZ,scadd

        LD (allbits+1),A ; reset all moves
        LD (bitset+1),A ; reset set bits

        INC (HL) ; increase multiplier

scadd   LD A,(HL)
        SUB 28
        LD (nrmoves+1),A
        LD E,A
scp     LD D,0
addpoint LD HL,score+3
        CALL addsc1
        DEC D
        JR NZ,addpoint
        DEC E
        JR NZ,scp
        POP AF
        POP DE
dokey   LD L,A ; save key pressed,code 1 less

        LD A,B
        SUB 18
        JR NZ,notramp

        LD (old+1),A ; reset current move
        LD (movesdone),A ; reset moves done

        LD A,(DE)
        CP 18
        PUSH HL
        CALL NZ,addscore ; each jump is a point
        POP HL

        EXX
        LD HL,(what+1)
        LD DE,mleft
        SBC HL,DE
        EXX
        JP NZ,eog ; men left on trampoline

        INC L ; test z pressed
        DEC L

        LD H,E
        JR NZ,less
        INC DE
        INC DE
less    DEC DE
        LD A,(DE)
        CP 10
        JR NC,playon
        LD E,H
playon  LD C,1
        XOR A
        LD (notramp+1),A
notramp LD A,L
        OR A

```

```

what      LD    HL,mleft
          PUSH BC
          PUSH DE
          LD    DE,screen
mkscr     LD    A,B
          ADD   A,A           ; each line double display
          LD    (DE),A
          INC   DE
          LD    A,(HL)
          LD    C,H           ; keep C > 1
          LDI
          OR    (HL)
          LDI
          DEC   B
          OR    A
          JR    NZ,mkscr

          LD    A,B
          LD    B,4
          CP    B
          JR    NC,settramp

          DEC   DE
          DEC   DE
          DEC   DE

settramp  LD    HL,tramp
settr     LD    A,B
          ADD   A,A
          LD    (DE),A
          INC   DE
          LDI
          LDI
          LD    L,tramp*256/256+2
          DJNZ settr
          POP   DE
          POP   BC

          LD    A,254
delay     LD    HL,frames     ; standard delay routine
          ADD   A,(HL)
wfr       CP    (HL)
          JR    NZ,wfr
          SUB   256-25
          JR    C,plloop
          LD    (frames),A     ; reset second counter

          LD    HL,time+3
          CALL  addsc1         ; add a second

plloop    JP    playloop

addscore  LD    HL,score+5
          DEFB #3A
ten       LD    (HL),28
sktime    DEC   HL
          LD    A,(HL)
          CP    14
          JR    Z,sktime
addsc1    INC   (HL)
          LD    A,L
          CP    time*256/256+2
          LD    A,38           ; test 10
          JR    NZ,test10

```

```

test10    LD    A,34                ; test 6
          CP    (HL)
          JR    Z,ten
          RET

space     EQU    #4400-$-150
          DEFS   space

x         EQU    101
lowres    DEFB   118
score     DEFB   28,28,28,28,28,0
compmove  DEFB   0,0,0,0

mulfact   DEFB   28,0
          DEFB   "T"+x,"R"+x,"A"+x,"M"+x,"P"+x,"O"+x
          DEFB   "L"+x,"I"+x,"N"+x,"E"+x,0

time      DEFB   28,14,28,28,0

hiscore   DEFB   28,28,28,35,31
          DEFB   118

screen    DEFB   255
init      LD     HL,#4020
ccline    DEC     L
          LD     (HL),B
          JR     NZ,ccline
          JP     start

vars      DEFB   128
?
last      EQU    $

```