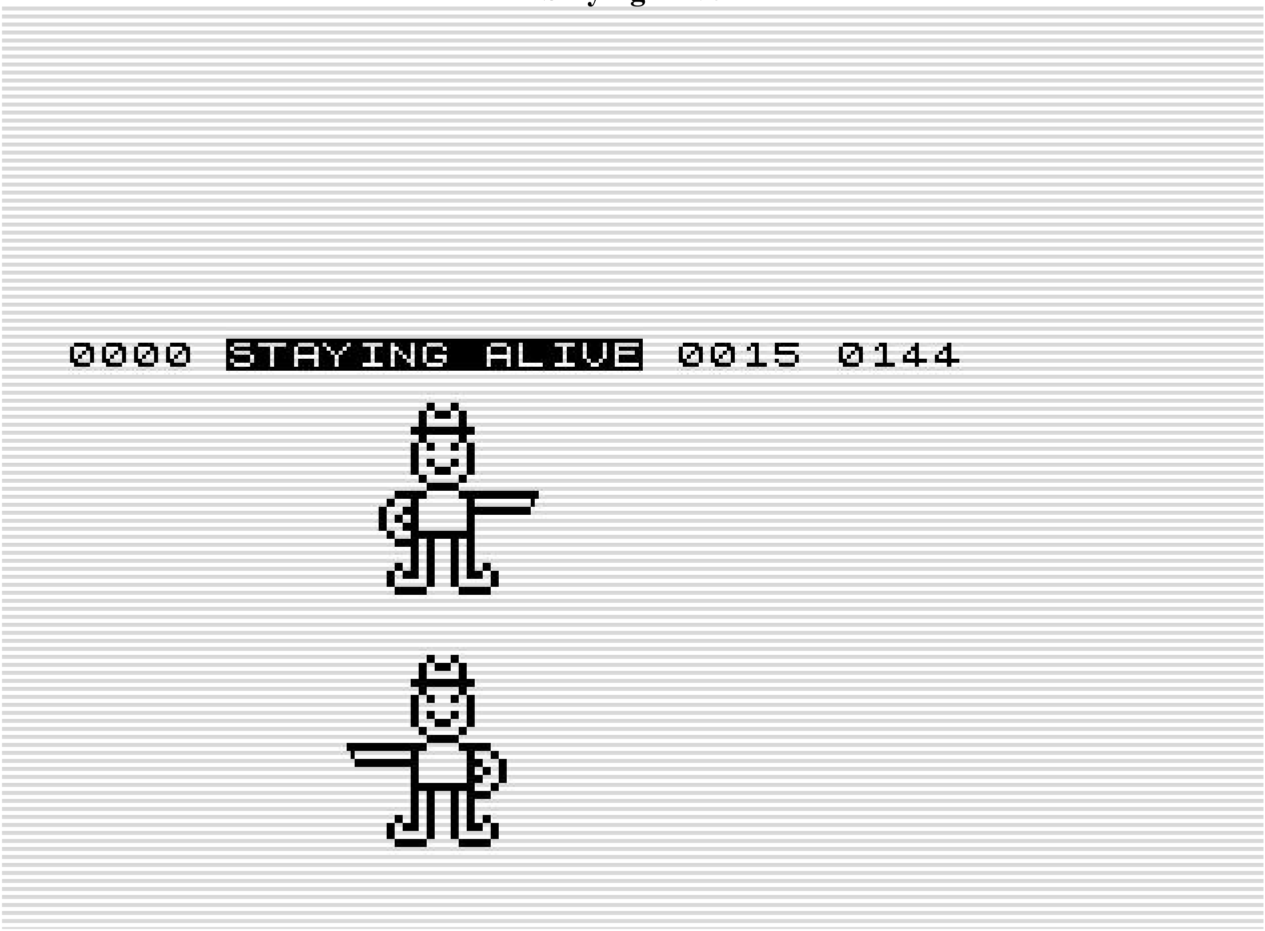
Staying Alive



```
0000 STAYING ALIVE 0015 0144
```

Ha Ha Ha Ha Staying alive... Staying alive.
Dance like John Travolta. Play 2 games in just 1K. First game is to repeat the growing dance like the game SIMON. Second game is do the move asap or run out of time.

```
; Staying Alive
; 2 games in 1K hires
; Game 1: a game like SIMON, can you dance like John Travolta
; Game 2: copy John Travolta's move within the time allowed

? * TORNADO *

           ORG  #4009              ;#4009
           DUMP 49161

basic      LD   H,#44              ; preset for SP set
           JR   init0              ; next step for initialization

           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             ; unused in this game, default

; 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

           LD   L,A              ; Make HL = #4400 for SP
           DEFB 17               ; LD DE,nn  ; skip taddr

taddr      DEFW 3213             ; used by ZX81
           EX   AF,AF'           ; delay intrupts
           DEFB #3A              ; LD A,(NN) ; skip frames

frames     DEFW 65535            ; used by ZX81
coords     JR   init             ; useable
prcc       DEFB 188              ; used by ZX81
sposn      DEFB 33,24            ; used by ZX81
cdflag     DEFB #40

init       LD   IX,hr            ; set HR
           LD   SP,HL            ; set SP
           JP   start            ; go to start
; initroutine also sets screen on correct startaddress

; the default screen is a still standing John Travolta
screen     DEFB 0,0,12,48,0,0
           DEFB 0,0,51,204,0,0
           DEFB 0,0,48,12,0,0
           DEFB 0,0,255,255,0,0
           DEFB 0,0,48,12,0,0
           DEFB 0,0,204,51,0,0
           DEFB 0,0,192,3,0,0
           DEFB 0,0,204,51,0,0
           DEFB 0,0,195,195,0,0
           DEFB 0,0,48,12,0,0
           DEFB 0,0,15,240,0,0

; here the handdisplay is swapped for display
hands      DEFB 0,15,240,15,240,0
           DEFB 0,48,192,3,12,0
           DEFB 0,195,192,3,195,0
           DEFB 0,204,192,3,51,0
           DEFB 0,195,192,3,195,0
           DEFB 0,48,255,255,12,0
           DEFB 0,15,204,51,240,0
           DEFB 0,0,204,51,0,0

; here the feetdisplay is swapped for display
feet       DEFB 0,0,204,51,0,0
           DEFB 0,12,204,51,48,0
           DEFB 0,51,204,51,204,0
           DEFB 0,48,12,48,12,0
           DEFB 0,15,240,15,240,0

; table to calculate correct swap address
tab        DEFB 0,3,feet-hands,feet-hands+3
           DEFB 0,lefthand-righthand
```

```
udg         DEFB rightfoot-righthand
            DEFB leftfoot-righthand

; the lowres top lines
hr          LD   HL,lowres+#8000    ; the lowres display
            LD   BC,#241            ; minimum needed
            LD   A,#1E
            LD   I,A
            LD   A,#FB
            CALL #2B5

            LD   A,(HL)             ; timing only

            CALL sw1                ; show John's current move
            CALL display            ; display John Travolta
            CALL sw1                ; undo John's current move

            LD   A,(HL)             ; timing

            CALL sw2                ; show players current move
            CALL display            ; display player
            CALL sw2                ; undo players current move

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

righthand   DEFB 255,255,240        ; right hand graphic
            DEFB 64,0,192
            DEFB 63,255,192
            DEFB 0,0,192
            DEFB 0,0,192
            DEFB 0,0,255
            DEFB 0,0,204

rightfoot   DEFB 0,243,12           ; right foot graphic
            DEFB 3,12,48
            DEFB 0,192,192
            DEFB 0,63,0
            DEFB 0,0,0

leftfoot    DEFB 48,207,0           ; left foot graphic
            DEFB 12,48,192
            DEFB 3,3,0
            DEFB 0,252,0
            DEFB 0,0,0

lefthand    DEFB 15,255,255         ; left hand graphic
            DEFB 3,0,2
            DEFB 3,255,252
            DEFB 3,0,0
            DEFB 3,0,0
            DEFB 255,0,0
            DEFB 51,0,0

rnd         LD   HL,(frames)        ; set a random start in ROM
            LD   A,R
            ADD  A,H
            LD   H,A
            RET

sw2         LD   HL,0               ; get player move to show
```

```
pos2        LD   DE,0              ; where to place on screen
            JR   swap
sw1         LD   HL,0              ; get John's move to show
pos1        LD   DE,0              ; where to place on screen
            LD   A,(HL)            ; timing
            LD   A,(HL)            ; timing

; swapping saves memory to store altered screen
swap        LD   B,7               ; 7 lines to swap
            LD   A,(HL)            ; timing
copy        LD   C,6               ; 3! bytes to swap
loop        LD   A,(DE)            ; get screenvalue
            LDI                    ; copy udg to screen
            DEC  HL
            LD   (HL),A            ; write screen to udg
            INC  HL
            DEC  C                 ; per byte 2 dec C 1x from LDI
            JR   NZ,loop           ; swap 3 bytes
            INC  DE                ; adjust screenline
            INC  DE
            INC  DE
            DJNZ copy              ; do all screenlines
            RET

display     LD   B,48              ; 48 screenlines in game 48 ;)

            LD   HL,screen         ; start of screen

again       PUSH HL                ; timing, 23 tstates delay
            INC  SP
            INC  SP

nline       LD   A,H               ; mostly delay
            LD   I,A               ; setttting I here saves 2 byte
            EX   (SP),HL
            EX   (SP),HL
            PUSH HL
            POP  HL
            LD   A,(HL)
            OR   A

            LD   DE,6

            LD   A,L               ; start of line
            DEC  B
            CALL lbuf+#8000
            LD   A,B
            AND  1
            JR   NZ,again          ; show each line 2x

            ADD  HL,DE             ; point to next line
            INC  B
            DJNZ nline             ; test end of screen
            RET

lbuf        LD   R,A               ; show 6 bytes
            DEFW 0,0,0             ; Upper memory has no
            RET                    ; 48K bug in this game

gameover    LD   HL,score-1        ; test for hiscore
            LD   BC,5
fhi         DEC  C
            JR   Z,start
            INC  HL
```

```
           INC  DE
           LD   A,(DE)
           CP   (HL)
           JR   Z,fhi
           CALL C,#19F9


start      LD   A,%11110111       ; read 1-5
           IN   A,(254)
           CPL
           AND  3                 ; 1 or 2 only
           JR   Z,start

           LD   B,A

           LD   HL,#1C1C          ; reset score
           LD   (score),HL
           LD   (score+2),HL

           CALL stand             ; reset moves old game

           LD   HL,0              ; reset stepcounter
           LD   (steps+1),HL

           CALL rnd
           LD   (dance1+2),HL     ; set start of moves

           DEC  B
           JP   Z,dance           ; game 1 start

; ROM-pointer is ok, score reset
; show move, wait time
; show player move, undo player move
; in time continue, otherwise game over
game2      LD   C,255-7           ; -8
nextmove   LD   B,10              ; 10 tries to to the move
           CALL nextstep          ; find the move

keyinp     LD   DE,pos1+2
           CALL showmove          ; show John's move

           LD   A,C               ; some delay
           CALL delay             ; speeds up after time

           CALL readkey           ; read keyboard
           JR   Z,checkmove       ; pressed key found
           DJNZ keyinp            ; do all loops
           JR   go2               ; alas too late gameover

checkmove  PUSH AF                ; save key pressed
           LD   DE,pos2+2         ; show your move
           CALL showmove+1
           LD   A,250             ; some delay
           CALL delay             ; to show move
           CALL stand             ; and back to rest
           POP  AF                ; get key
           XOR  (HL)              ; test against John's move
           AND  3                 ; only 2 bits needed
go2        LD   DE,hi2-1          ; preset hiscore address
           JR   NZ,gameover       ; not the same, game over
           PUSH HL                ; save ROM-pointer
setpoint   CALL addscore          ; add a point
           LD   A,L
           CP   score*256/256+2   ; test 100 points altered
           JR   NC,nospeed        ; if not, same speed
           LD   A,C
           ADD  A,3
           JR   C,nospeed         ; test minimun speed
           INC  C                 ; not reached, speed up

nospeed    DJNZ setpoint          ; add all the points
           POP  HL                ; get ROM-pointer
           JR   nextmove          ; do next move

; game 1 is here
dance      XOR  A                 ; first show the moves
           LD   (inptest+1),A

steps      LD   BC,0
           INC  BC
           LD   (steps+1),BC      ; add a move to the dance

dance1     PUSH BC                ; save moves
           LD   HL,0
           LD   A,48              ; nr > 3, this is game 48
           LD   (old+1),A         ; make last key different

dance2     CALL nextstep

inptest    LD   A,0               ; showing or replaying?
           LD   (t2+1),A          ; save for later
           OR   A
           PUSH AF                ; save result for score
           JR   Z,skipinp         ; showing

; read input and translate to show
reread     CALL readkey
           JR   NZ,reread         ; wait for valid key pressed
           AND  3                 ; 2 bits only needed

           PUSH AF                ; save keycode

           LD   DE,pos2+2         ; set player move
           CALL playmove

           POP  AF                ; get keycode
           XOR  (HL)              ; test against John's move
           AND  3
           LD   (okmove+1),A      ; save result

skipinp    LD   DE,pos1+2         ; show John's move
           CALL showmove
           CALL delay-2           ; move some time visible
           POP  AF                ; get show/input again
           JR   Z,skipscore       ; show, no score

okmove     LD   A,0               ; was the move valid?
           OR   A

           LD   DE,hi1-1          ; preset possible hiscore
           JP   NZ,gameover       ; false move

           PUSH HL                ; save ROM pointer

           CALL addscore          ; get a point

           POP  HL                ; get ROM pointer
```

```
skipscore  DEC  BC                 ; another move done
           LD   A,B
           OR   C                  ; full dance done?
           JR   NZ,dance2          ; play moves

           INC  A                  ; set
           LD   (inptest+1),A

           CALL stand              ; show no moving dancers

t2         LD   A,0                ; test player dance done
           OR   A
           POP  BC
           JR   Z,dance1           ; do full replay of dance

           JR   dance              ; add a step

addscore   LD   HL,score+4
           DEFB 17
addten     LD   (HL),28
           DEC  HL
           INC  (HL)
           LD   A,(HL)
           CP   38
           JR   Z,addten
           RET


nextstep   INC  HL                 ; goto next move
           LD   A,H
           AND  #1F
           LD   H,A                ; but keep it in ROM

           LD   A,(HL)             ; get move
           AND  3                  ; 2 bits only
old        CP   0                  ; test against previous move
           JR   Z,nextstep         ; move must be different
           LD   (old+1),A          ; save new move for test
           RET

readkey    PUSH BC
           LD   A,(lastk)          ; get in port
           EX   DE,HL
           LD   HL,keytab
           LD   BC,5
           CPIR                    ; seek in keytable
           EX   DE,HL
           LD   A,C                ; position in A
           POP  BC
           RET

keytab     DEFB %11111011          ; right arm
           DEFB %01111111          ; left foot
           DEFB %11111110          ; right foot
           DEFB %11011111          ; left arm

showmove   LD   A,(HL)             ; get ROM-value
           AND  3                  ; 2 bits only
playmove   PUSH BC                 ; save steps
           PUSH HL                 ; save ROM-pointer
           LD   H,tab/256          ; translate code
           ADD  A,tab*256/256      ; to position on screen
           LD   L,A
           LD   A,(HL)             ; get displacememt from hands
           INC  HL                ; go to next table
           INC  HL
           INC  HL
           INC  HL
           LD   B,(HL)            ; get udg-pointer

           LD   HL,hands
           ADD  A,L
           LD   L,A               ; HL now correct screenpointer

           EX   DE,HL
           LD   (HL),D            ; write screenpointer to
           DEC  HL                ; displayroutine
           LD   (HL),E

           LD   DE,righthand
           LD   A,E
           ADD  A,B
           LD   E,A               ; DE now correct UDG

           DEC  HL
           DEC  HL

           LD   (HL),D            ; write UDGpointer to
           DEC  HL                ; displayroutine
           LD   (HL),E

           POP  HL                ; retrieve ROM-pointer
           POP  BC                ; retrieve steps
           RET

stand      XOR  A                 ; swap ROM with ROM
           LD   (sw1+2),A         ; this will do nothing
           LD   (pos1+2),A        ; and therefore
           LD   (sw2+2),A         ; showing the default screen
           LD   (pos2+2),A        ; for timing swap is needed

           LD   A,255-24          ; -25
delay      PUSH HL
           LD   HL,frames
           ADD  A,(HL)
wfr        CP   (HL)
           JR   NZ,wfr
           POP  HL
           RET

x          EQU  101

lowres     DEFB 118
score      DEFB 28,28,28,28,0
           DEFB "S"+x,"T"+x,"A"+x,"Y"+x,"I"+x,"N"+x,"G"+x
           DEFB 128,"A"+x,"L"+x,"I"+x,"V"+x,"E"+x,0
hi1        DEFB 28,28,28,28,0
hi2        DEFB 28,28,28,28
           DEFB 118

vars       DEFB 128
?
last       EQU  $
```