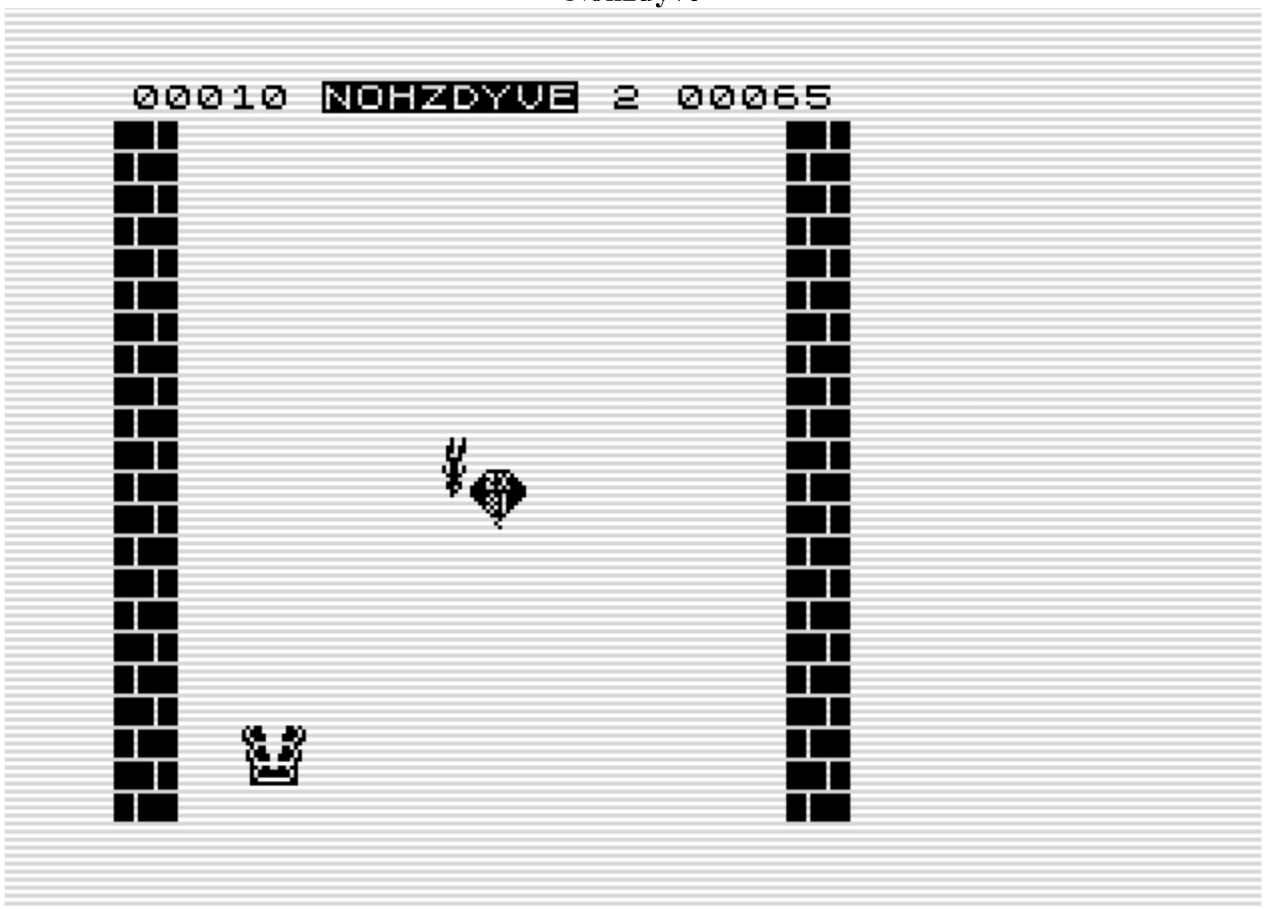
Nohzdyve



The challenge of Nohzdyve was to add the walls to the display. 3 characters in a row is not difficult. Adding the walls will look as if 7 characters are drawn to the screen while screen is still 23 characters wide. A record.... and a trick.

```
; Nohzdyve
; Game 65 in 1K hires for the ZX81

? * TORNADO *

            ORG  #4009              ;#4009
            DUMP 49161

lastline    EQU  #43F8
lastflag    EQU  #43FE

xposplay    EQU  #43B5

basic       LD   D,#C0              ; 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 during loading
margin     DEFB 55                ; used by ZX81
nxtlin     DEFW basic             ; reusable after load

init0      LD   E,L               ; DE = HL + #8000
           DEFB #26               ; LD H,64
flagx      DEFB 64

           XOR  A
           EX   AF,AF'

taddr      DEFW 0                 ; used by ZX81
           LD   B,4               ; 1K code to copy

frames     DEFW #DD*256+1         ; used by ZX81

coprcc     LD   HL,hr             ; set IX to HR
sposn      JR   init1

cdflag     DEFB 64                ; used by ZX81

copyline   XOR  A
           JR   erline

           DEFB 251               ; first line data, left wall
line2      DEFB 255,223           ; second line data, right wall

erline     LD   HL,line2-1        ; begin of line1
           LD   DE,#4000          ; copy to start of RAM
           LD   C,23
           LDIR                   ; copy line over sysvar
           LD   E,2               ; first address to clear
           LD   L,erline*256/256  ; first on line2
           LD   B,20
           JR   erase             ; continu out of line
           DEFB 251               ; right wall line1
           DEFB 255,223           ; right wall line2

erase      LD   (HL),A            ; clear line
           LDI                    ; copy clear, 1x DEC B C<0
           DJNZ erase             ; clear 19 fields

; through end of game to start of new game
\eog        LD   HL,score-1        ; test score
           LD   DE,hi-1           ; against hiscore
           LD   BC,6              ; score has 5 digits, 6 tested
same       INC  HL                ; goto next digit score
           INC  DE                ; goto next digit hiscore
           DEC  C                 ; 1 digit less to copy
           LD   A,(DE)            ; When C=0 (DE) always > (HL)
           CP   (HL)              ; so NEVER hiscore when sc=hi
           JR   Z,same            ; still same score
           CALL C,#19F9           ; set new hi with LDIR in ROM

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

```
            LD   B,5
            LD   HL,score
ressc       LD   (HL),28           ; clear an old score
            INC  HL
            DJNZ ressc
            LD   HL,lives          ; set "4" lives
            LD   (HL),32

liveless    LD   HL,lives
            DEC  (HL)              ; 1 live lost
            LD   A,28
            CP   (HL)
            JR   Z,eog             ; test game over

            LD   HL,dispstack+6    ; repair dispstack for
            LD   (HL),empty12*256/256
            LD   L,dispstack*256/256
            LD   (HL),empty11*256/256
            LD   DE,dispstack+12   ; a new game. Also CLS
            LD   BC,120            ; since first line will
            LDIR                   ; NEVER hold a graphic

            LD   A,10              ; show falling person line1
            LD   (xposplay),A
            XOR  64
            LD   (xposplay+6),A    ; show line2

waitspace   LD   A,(lastk)
            LD   (movekey+1),A     ; signal no direction
            SUB  127
            JR   NZ,waitspace      ; wait for space for next life
            LD   (dead+1),A        ; reset dead

play        CALL hittest           ; test hit after move player
            LD   A,(dispstack+5)
            OR   A
            JR   NZ,play3          ; cheak dead each 2nd loop so
dead        OR   0                 ; player is swapped to normal
            JR   NZ,liveless       ; dead

play3       LD   HL,xposplay
            LD   A,(HL)            ; get line1
            LD   (HL),25           ; set display player off
            PUSH AF
            LD   A,(HL)
            LD   (xposplay+6),A    ; set 2nd display off
            LD   DE,dispstack
            LD   HL,dispstack+6
            LD   BC,6*21
            LDIR                   ; move screen 1 line up

            XOR  A
            LD   (lastline),A      ; erase moved up end of screen

            DEC  HL
            LD   A,(HL)
            XOR  64
            LD   (HL),A            ; swap displaystone last line

            LD   DE,dxtab
            PUSH DE
            LD   HL,dxtab+1
            LD   C,21
            LDIR                   ; move dx also 1 line up
```

```
            POP  HL                ; get dx table
            LD   DE,dispstack+2
            LD   B,22
            DEFB #3A               ; hide direction swap
swapdir     XOR  (HL)
            LD   (HL),A
again       LD   A,(DE)            ; get xpos graphic
            ADD  A,(HL)            ; add dx, 1 too much so
            DEC  A                 ; it can be stored in lbuf
            LD   C,A               ; save result
            AND  191               ; take off display bit
            SUB  2
            CP   18                ; test wall hit
            LD   A,2               ; preset swap direction
            JR   NC,swapdir        ; swap dir and do new dx
            LD   A,C               ; get result
            LD   (DE),A            ; write result
            INC  HL                ; goto next dx
            LD   A,E
            ADD  A,6
            LD   E,A               ; goto next line
            DJNZ again             ; do all lines

; large graphics go up 1 line, but falling graphic stays same
; display is hidden in large graphics. Each line up
; the falling graphic must swap top/bottom for right display
; during swap display of player must be off.
            LD   HL,swaptab-1      ; the graphic table
            LD   D,H
            LD   C,9*3             ; 3 udg, 1 out swap 8 in swap
alludg      INC  HL
            LD   E,(HL)            ; top part in large graphic
            INC  HL
            PUSH HL                ; save table
            LD   L,(HL)            ; bottom in large graphic

            LD   B,8
swap        INC  HL                ; skip large graphic
            INC  HL
            INC  DE                ; skip large graphic
            INC  DE
            LD   A,(DE)            ; get current top
            LDI                    ; move bottom to top
            DEC  HL                ; undo INC HL from LDI
            LD   (HL),A            ; move to to bottom
            INC  HL                ; sync HL with DE
            DJNZ swap              ; swap full graphic
            POP  HL                ; get graphicpointer
            DEC  C
            JR   NZ,alludg         ; swap all falling graphic

            POP  DE                ; xpos to d
            LD   E,D               ; save xpos
            PUSH DE
            CALL hittest2          ; test hit after move graphics
            POP  DE
            LD   C,2               ; preset right
            LD   A,%11011111
            IN   A,(254)
            RRA
            JR   NC,cfnd           ; right pressed
            DEC  C                 ; preset left
            RRA                    ; only carry needed, so RRA will do here!
```

```
            JR   NC,cfnd           ; left pressed
movekey     LD   C,0               ; get old direction
cfnd        LD   A,C
            LD   (movekey+1),A     ; save direction as old
            DEC  C
            JR   NZ,right
            DEC  D
right       DEC  C
            JR   NZ,testvalid
            INC  D
testvalid   LD   A,D
            AND  191               ; take of display bit
            SUB  2
            CP   19                ; test wall hit
            JR   C,valid
            LD   (dead+1),A        ; signal dead
            LD   D,E               ; undo move
valid       LD   A,D

            LD   (xposplay+6),A    ; save xpos player
            XOR  64
            LD   (xposplay),A      ; on both lines

cnt         LD   A,0
            DEC  A                 ; change counter
            AND  7                 ; also reset carry
            LD   (cnt+1),A         ; save counter
            RR   A                 ; Don't use RRA, Z-flag needed
                                   ; luckily I knew this
            LD   A,swaptab*256/256 ; default empty
            JR   NZ,afnd           ; move in space

; place a new graphic at the bottom
            JR   NC,getudg         ; set part 2 of same graphic

            CALL rnd               ; get a graphic
            ADD  A,A               ; table has double values
            ADD  A,swaptab*256/256-2 ; we have a new graphic
            LD   (getudg+1),A      ; save graphic for part 2

            LD   E,18
            CALL rnd+2
            ADD  A,65
            LD   (lastline-2),A    ; set random start
            XOR  64
            LD   (lastline+4),A    ; also on second line
            CALL rnd               ; A=1, 2 or 3
            DEC  A                 ; A = 0 1 or 2
            AND  2                 ; A - 0 or 2
            LD   (dxtabend),A      ; set random dx left/right
            SCF                    ; set carry needed

getudg      LD   A,0               ; get current graphic part 2
afnd        ADC  A,0               ; get right pointer
            LD   L,A
            LD   H,swaptab/256
            LD   L,(HL)
            LD   (lastline-4),HL   ; write udg

; now repair correct xpos
            LD   HL,lastline-2
            LD   A,(HL)
            XOR  64
            LD   (HL),A
```

```
            LD   HL,frames         ; delay to show screen
            LD   A,(HL)
            SUB  4
wfr         CP   (HL)
            JR   NZ,wfr
            JP   play

hittest     LD   A,(xposplay)
            LD   D,A
hittest2    LD   A,(xposplay-1)    ; get x udg
            CP   D                 ; test against xplayer
            JR   Z,d2test
            INC  A                 ; graphic has 2 positions
            CP   D                 ; test 2nd pos against xplayer
            RET  NZ
d2test      LD   A,(xposplay-3)    ; test space or balloon
            CP   udg21*256/256
            JR   NC,testother      ; balloon captured

; change balloon into spaces
            LD   HL,empty11
            LD   DE,empty12
            LD   A,(xposplay+2)
            BIT  6,A
            JR   Z,setempty
            EX   DE,HL
setempty    LD   (xposplay+3),DE   ; replace balloom with space
            LD   (xposplay-3),HL
            LD   (xposplay-9),HL

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

testother   CP   empty11*256/256   ; space hit, continue
            RET  NC
            LD   (dead+1),A        ; signal dead on graphic
            RET

repline     LD   HL,#4002          ; movement sets final row
            LD   DE,#4042          ; so we clear it
            LD   B,19              ; in hr solved would cost
cllast      XOR  A                 ; a display position
            LD   (HL),A            ; now 23 otherwise 22
            LD   (DE),A
            INC  DE                ; line is set here out of hr
            INC  HL                ; routine or program would not
            DJNZ cllast            ; fit memory
            JP   #2A4

space       EQU  #4206-$
            DEFS space

swaptab     DEFB empty12*256/256
            DEFB empty11*256/256
            DEFB udg12*256/256
            DEFB udg11*256/256
```

```
            DEFB udg22*256/256
            DEFB udg21*256/256

udg11       DEFB 7,224,9            ; the balloon with diver
            DEFB 9,80,9
            DEFB 29,184,27
            DEFB 59,188,27
            DEFB 121,94,18
            DEFB 127,254,30
            DEFB 125,190,12
            DEFB 58,188,45
udg12       DEFB 29,184,45
            DEFB 10,176,30
            DEFB 5,160,12
            DEFB 3,192,30
            DEFB 0,128,26
            DEFB 1,0,12
            DEFB 0,128,8
            DEFB 0,0,0

udg21       DEFB 88,26,9            ; the teeth with diver
            DEFB 184,29,9
            DEFB 188,61,27
            DEFB 220,59,27
            DEFB 96,6,18
            DEFB 44,52,30
            DEFB 92,58,12
            DEFB 94,122,45
udg22       DEFB 110,118,45
            DEFB 32,4,30
            DEFB 54,108,12
            DEFB 47,244,30
            DEFB 47,244,26
            DEFB 48,12,12
            DEFB 63,252,8
            DEFB 0,0,0

empty11     DEFB 0,0,9              ; space with diver
            DEFB 0,0,9
            DEFB 0,0,27
            DEFB 0,0,27
            DEFB 0,0,18
            DEFB 0,0,30
            DEFB 0,0,12
            DEFB 0,0,45
empty12     DEFB 0,0,45
            DEFB 0,0
lbuf        DEFB 30                 ; show 19 columns, no walls
            DEFB 0,0,12
            DEFB 0,0,30
            DEFB 0,0,26
            DEFB 0,0,12
            DEFB 0,0,8
            DEFB 0,0,0
            JP   NC,cloop           ; after final line end
            JP   savesp             ; with carry set

rnd         LD   E,3
            LD   A,(frames)
rseed       ADD  A,1
            LD   D,A
            RRCA
            RRCA
            RRCA
```

```
            XOR  31
            ADD  A,D
            SBC  A,255
            LD   (rseed+1),A
sube        SUB  E
            JR   NC,sube
            ADC  A,E
            RET

hr          LD   HL,lowres+#8000   ; the lowres display
            LD   BC,#201           ; minimum needed
            LD   A,#1E
            LD   I,A
            LD   A,#F9
            CALL #2B5

            DEC  HL                ; sync hires with lowres
            NOP

            LD   D,#40             ; display lines in #40..
            LD   A,D
            LD   I,A               ; set highbyte for display

            EXX

            LD   HL,(xposplay)     ; get xpos player
            LD   H,A               ; set correct highbyte

            LD   IX,lbuf+#8000     ; JP (IX) is 8 tstate

            LD   B,7               ; preset loopcounter
            LD   IY,lbuf2+#8000    ; JP (IY) is 8 byte

            LD   (savesp+1),SP     ; we use another stack
            LD   SP,dispstack      ; set new displaystack

bloop       LD   C,B               ; set loopcounter
            EXX
            POP  HL                ; graphicspointer
            POP  BC                ; xpos player and second graph

            LD   E,C               ; xpos udg
            LDI                    ; copu udg to line

            LD   C,E               ; E is C+1
            LDI                    ; C now again ok, copy UDG

            LD   E,B               ; get xpos player
            LD   A,(HL)            ; get udg player
            LD   (DE),A            ; write to screen
            POP  AF                ; get displaypointer and flag
            LD   R,A               ; set display
            JP   (IX)              ; R+2 for display

; fixed end of HR-routine
savesp      LD   SP,0              ; get old stackpointer
            LD   IY,#4000          ; repair IY
            CALL #292              ; back from intrupt
            CALL #220
            LD   IX,hr             ; set for next display
            JP   repline           ; repair line set by bloop

cloop       INC  L                 ; point to next UDG data
            EXX
```

```
           LD   (HL),0            ; repair first line player
                                  ; other UDG with empty UDG

c2loop     EXX                    ; now back to UDGpointers
           LD   E,C               ; set X floating UDG
           LDI                    ; copy data
           LD   C,E               ; again E=C+1 trick
           LDI                    ; copy second data
           LD   E,B               ; set X player
           INC  BC                ; repair C now with INC
           LDI                    ; copy data
           EXX                    ; swap to counter
           DEC  C                 ; decrease UDG counter
           JP   (IY)              ; do display

lbuf2      LD   R,A               ; now show more fields
dxtab      DEFW 0,0,0,0,0,0,0,0   ; show 23 columns, walls incl.
           DEFB 0,0,0,0,0,0       ; also used as dx-table
dxtabend   DEFB 0
           JP   Z,bloop           ; go to next line
           JP   c2loop            ; do full line

; the low and hires screens
x          EQU  101               ; text inverted

lowres     DEFB 118
score      DEFB 28,28,28,28,28,0
           DEFB "N"+x,"O"+x,"H"+x,"Z"+x,"D"+x,"Y"+x
           DEFB "V"+x,"E"+x,0
lives      DEFB 31,0
hi         DEFB 28,28,28,34,33
           DEFB 118

; from builtstack to dispstack 32 bytes SP room
; on init the screen is built in SP memory
builtstack LDIR                   ; piece of stack used on setup
           LD   A,1
           LD   (lastflag),A
           JP   copyline

           DEFS 22                ; full stack 32 bytes

spc        EQU  #4376-$
           DEFS spc

; The displaystack is created here on startup.
dispstack  DEFW empty11           ; udg
           DEFB 4,25              ; posx udg and player
           DEFB 0,0               ; flag and displayline1
           DEFW empty12
           DEFB 68,25
           DEFB 0,64              ; displayline2

init       LDIR                   ; repair 48K bug
           LD   HL,dispstack      ; preset registers
           LD   SP,HL             ; move stackpointer
           LD   DE,dispstack+12
           LD   C,126             ; size of screenstack
           JP   builtstack

vars       DEFB 128
?
last       EQU  $
```