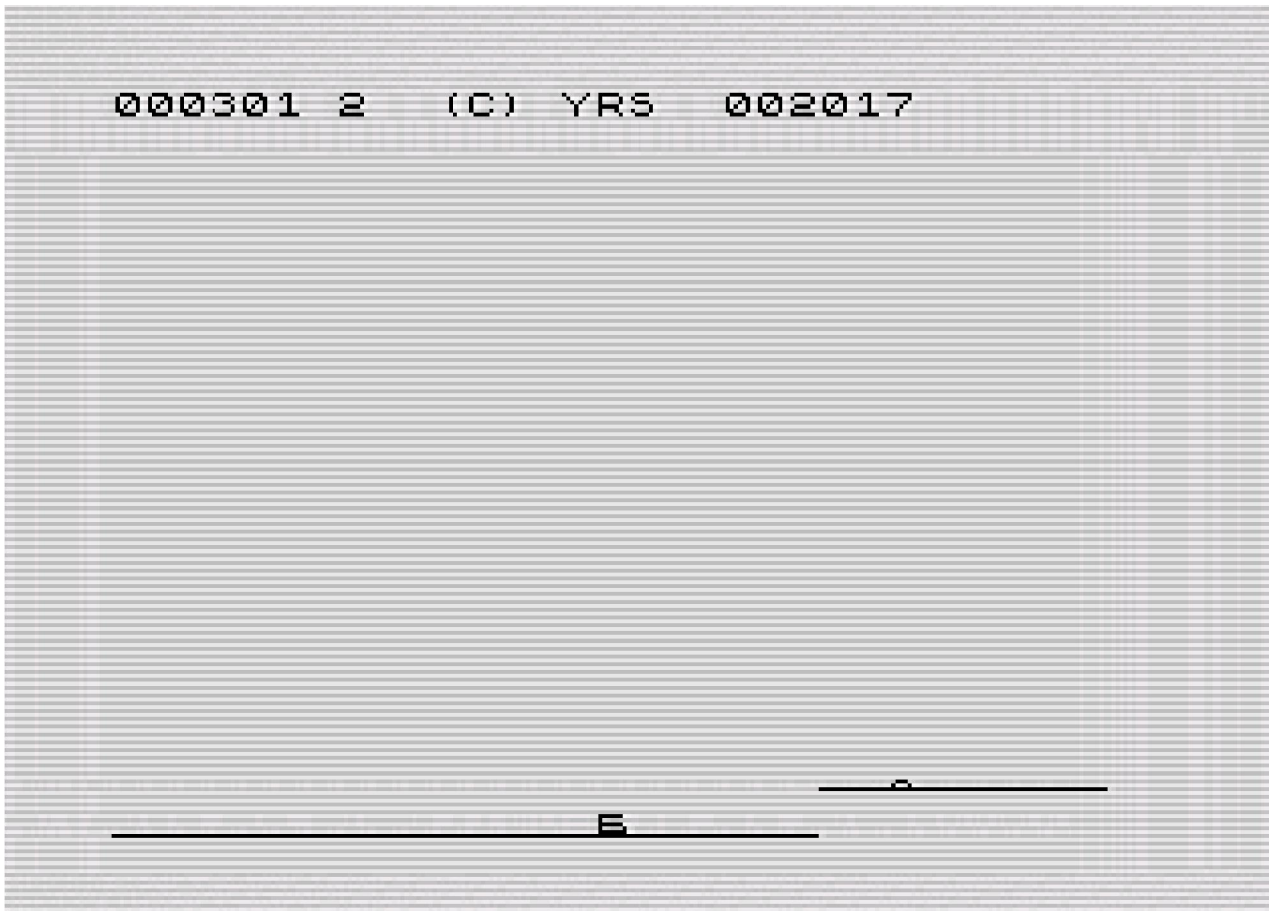


Iron ZXlider



Each 10th game must be a special.

This 30th game is a request. Can the online game Vector Runner be coded in 1K hires?

The base display of Blocky could be used, however the displayorder is not fixed, a new line can appear above the old. A new method was needed. After many tests this worked. Not swapping the lines (at lost of speed), but building an index with the lines in order.

The maximum of 11 displaylines is set in 6 full lines and 5 1 byte lines for the Zxlider-UDG.

This was needed to keep the entire screen within 256 bytes.

```
; Iron ZXlider

; screen
; #4000 screenindextable 18 bytes (y, index 43xx) * 9
; #4014 iron-UDG
; #4300-#43BF datatable 192 bytes (y, 31x screen) * 6
; #43C1 playertabel      10 bytes (y, character ) * 5

? * TORNADO *

                ORG   #4009                ;#4009
                DUMP  49161

disp            EQU   16

                JP     init
d_file          DEFW  dfile
dfcc            DEFW  dfile+1
```

```

var      DEFW vars
dest     DEFW 0
eline    DEFW last
chadd    DEFW last-1
xptr     DEFW 0
stkbot   DEFW last
stkend   DEFW last
berg     DEFB 0
mem       DEFW 0                ; not needed without fp
         DEFB 128
dfsz     DEFB 2
stop     DEFW 1
lastk    DEFB 255,255,255
margin   DEFB 55
wrinkles DEFW basic            ; wrinkle UDG's copied here
         DEFW 0                ; oldppc
         DEFB 0
strlen   DEFW 0
taddr    DEFW 3213
seed     DEFW 0
frames   DEFW 65535
coords   DEFB 0,0
prcc     DEFB 188
sposn    DEFB 33,24
cdflag   DEFB 64

eog       LD   HL,score-1        ; hiscore test
         LD   DE,hisc-1
         LD   BC,7
findhi    DEC   C
         JR   Z,begin
         INC  DE
         INC  HL
         LD   A,(DE)
         CP   (HL)
         JR   Z,findhi          ; test hiscore
         JR   NC,begin
         LDIR

begin      XOR   A
         LD   (lineidx+1),A      ; reset start line
         INC  A
         LD   (shiftcnt+1),A     ; 1 shift needed initially

wkey       LD   A,%10111111
         IN   A,(254)
         RRA
         JR   C,wkey            ; wait NewLine

         XOR   A
         LD   (#4000),A         ; "empty" screenindex
         LD   HL,1001
         LD   (speeddelay+1),HL ; undo increased speed

cls        LD   HL,#43BF
         LD   (HL),255          ; fake end of line initially needed
cls1       DEC   L
         LD   (HL),A            ; clear rest of screen
         JR   NZ,cls1

         LD   HL,score+7        ; reset score, set lives
         LD   B,6
         LD   A,28
         LD   (HL),34           ; set "5" lives ("6"-1)

```

```

ressc      DEC HL          ; skip space
           DEC HL
           LD (HL),A       ; erase score
           DJNZ ressc

           RRCA
           LD (frloop+1),A ; first 14 lines no turbulence

dead        LD HL,lifecnt   ; lost a live
           DEC (HL)
           LD A,(HL)
           SUB 28           ; is it "0"?
           JR Z,eog        ; end of game

           LD A,(#4000)     ; fetch highest line
           OR A
           JR NZ,add6      ; during game, set just above line
           LD A,52         ; add drop to line
add6        ADD A,7         ; drop time, size of udg and some space
           LD (irony+1),A  ; set Y of UDG iron

; erase iron-UDG
lptst      LD HL,turbolf   ; undo display TURBULENCE
           LD (HL),118

           LD HL,screen+160+disp
erairon     LD A,(HL)
           AND %11000000
           SUB %01000000
           JR NZ,noerase   ; only iron UDG gives zero
           LD (HL),A       ; erase UDG
noerase     LD A,L
           SUB 32
           LD L,A
           JR NC,erairon

; move iron, drop or jump
           LD L,192
droptest    LD A,(irony+1)
           LD E,A
           SUB 5
           CP (HL)
           JR Z,online     ; test UDG on a visible line

           LD A,L
           SUB 32
           LD L,A
           JR NC,droptest
           JR dodrop       ; in the air, thus drop

online      LD A,L         ; is there a line?
           ADD A,disp
           LD L,A
           LD A,(HL)
           INC A
           JR NZ,dodrop    ; on line, no drop

; read jump key
           LD A,(lastk)
           INC A
           JR Z,jpcnt      ; no key pressed

           LD A,18         ; set jump counter
           LD (jpcnt+1),A

```

```

dodrop    LD    A,(jpcnt+1)      ; still jumping?
          OR    A
          JR    NZ,jpcnt
          LD    A,E
          DEC   A                ; drop
          LD    (irony+1),A
          JR    Z,dead           ; fell out of screen

jpcnt     LD    A,0              ; looks like double code with dodrop.
          OR    A                ; a JR just above will save 3 bytes
          JR    Z,nojump         ; no need to change it
          DEC   A
          LD    (jpcnt+1),A
          LD    HL,irony+1       ; go up 1 line in jump mode
          INC   (HL)

nojump    LD    HL,score+5
          LD    D,(HL)
          CALL  addscore         ; 1 moving point scored
          LD    A,D
          AND   1
          JP    NZ,biron         ; built iron, 1/2 shift is done
          LD    HL,screen

shiftcnt  LD    D,0              ; how mny shifts until end of line
          LD    C,6              ; 6 lines (2 for wrinkle, 1 for line)*2
shift     LD    A,(HL)           ; check linenr
          INC   HL               ; skip y-pos
          LD    B,30             ; copy 30 bytes linedata
shl       INC   HL
          LD    A,(HL)           ; fetch character
          DEC   HL               ; move line
          LD    (HL),A           ; set character
          INC   HL
          DJNZ  shl
          LD    (HL),B           ; clear last position on line
          INC   A                ; is it a line?
          JP    NZ,noline        ; you moved items, not a line

          DEC   D                ; decrease shiftcounter
          LD    A,D
          LD    (shiftcnt+1),A   ; save shiftcounter

          LD    (HL),255         ; make line longer
          JR    Z,noobst         ; not at end of line, obstacle allowed

          AND   15               ; each 16th pos an "obstacle"
          JR    NZ,noobst        ; set no obstacle

          CALL  rnd
          AND   3                ; 0-3
          JR    Z,noobst         ; 0, obstacle is nothing
          PUSH  DE               ; save shiftcounter
          LD    DE,wrinkles+2    ; preset UDG
          DEC   A
          JR    Z,highwrin       ; 1, high wrinkle
          DEC   E                ; 2 and 3 low wrinkle
          DEC   E                ; correctpointer to low wrinkle
highwrin  PUSH  HL               ; save current line
setobst   LD    A,L
          SUB   32
          LD    L,A              ; obstacle 32 bytes above line
          LD    A,(DE)

```

```

        INC     E
        LD      (HL),A
        LD      A,E
        AND     1                ; obstacle is 2 bytes, use odd-number test
        JR      Z,setobst
        POP     HL                ; retrieve line
        POP     DE                ; retrieve shiftcounter

noobst   XOR     A
        OR      D
        JR      NZ,noline        ; not end reached
        LD      (HL),D           ; end reached, erase

; add new line
findline LD      A,(lineidx+1)    ; fetch current line
        AND     A                ; test start of game
        LD      A,7              ; set fixed startvalue
        CALL    NZ,rnd           ; otherwise change height of line
        AND     7
        SUB     3                ; +/- 3
        JR      Z,falsernd       ; it must change a line
        LD      D,A              ; not needed anymore after small change
        ADD     A,A              ; 6
        ADD     A,A              ; 12
lineidx  ADD     A,0              ; add old
falsernd LD      D,A              ; save old A and set in D
        SUB     3                ; a block of 3 needed
        CP      150              ; test <=0, >150 or dy=0
        LD      A,D              ; restore old A
        JR      NC,findline      ; out of range or dy = 0
lfind    LD      (lineidx+1),A    ; base for new line
        LD      A,L              ; or other way around
        CP      128
        SBC     A,A
        AND     96
        LD      L,A              ; new start calculated

; current new line position
        LD      B,3
linenr   LD      (HL),D           ; set 3 lines in screen memory
        DEC     D
        LD      A,L
        ADD     A,32
        LD      L,A
        LD      C,B              ; to end noline loop
        DJNZ    linenr

        DEC     L
        LD      (HL),255         ; set start of line

        CALL    rnd
        AND     63                ; length 0-63
        ADD     A,32              ; length 32-95, min length 32
        LD      (shiftcnt+1),A    ; reset shiftcounter

        LD      A,(frloop+1)      ; a new line, decrease counter
        DEC     A                ; to turbulence
        JR      NZ,setfrlp

turbtime LD      A,0              ; No we have turbulence
        DEC     A
        AND     3
        LD      (turbtime+1),A    ; duration is 4 full lines
        JR      NZ,noline        ; still turbulence

```

```

        LD      (turbsh+1),A      ; undo display
        CALL    rnd
        AND     7                  ; new delay turbulence
        OR      8                  ; at least 8 lines free

setfrlp   LD      (frloop+1),A

noline    INC     HL              ; point to start of next line
          DEC     C                ; do a full screen
          JP      NZ,shift

; built iron
biron     LD      DE,#4014
irony     LD      BC,#C100        ; Y-coordinate and table iron
ibloop    LD      HL,screen+160   ; start at last line on screen
setiron   LD      A,C
          CP      (HL)            ; on this line?
          LD      A,L
          JR      NZ,scr2        ; no, continue

scr1       ADD     A,disp          ; set on scr1, the linescreen
          LD      L,A             ; halfway the line
          LD      A,(HL)
          CP      %00011000      ; low wrinkle hit
          PUSH    HL
          LD      HL,score+4      ; 10 pos
          CALL    Z,addscore      ; a score and reset of A
          POP     HL
          OR      A
          JR      Z,ironend       ; place on line and handled wrinkle
          INC     A               ; jump through a line?
          JP      NZ,dead        ; high wrinkle hit

scr2       SUB     32             ; test full scr1
          LD      L,A
          JR      NC,setiron

          LD      L,B             ; not on scr1, set on scr2
          LD      (HL),C         ; set current y
          INC     L
          INC     L
          LD      (HL),A         ; next line no show indicator
          LD      B,L            ; save for next UDG-position
          DEC     L              ; set next pointer
ironend    LD      A,(DE)         ; iron udg
          LD      (HL),A         ; set on correct screen
          LD      A,(DE)         ; fetch again, redundant code
skipudg    INC     DE             ; point to next udg
          DEC     C               ; Y 1 down
          CP      %01111111      ; test on bottom of UDG
          JR      NZ,ibloop      ; set full UDG

; shifted, now built screen for display
builtscr   LD      DE,#4360      ; "second" screen line
          LD      HL,#4300      ; "first" screen line, lines swap in order
          LD      A,(DE)         ; check order
          CP      (HL)
          JR      C,noswap       ; highest in HL
          EX      DE,HL          ; lines swapped
noswap     PUSH    DE            ; save lowest
          PUSH    DE            ; 2x, POP is twice
          LD      DE,#4000      ; screenindex
          LD      B,3
scrsort    LD      A,(HL)        ; fetch highest

```

```

LD      (DE),A          ; set inscr
LD      A,L
INC     A                ; fetch index
INC     DE
LD      (DE),A          ; set indexpointer
INC     DE
ADD     A,31
LD      L,A              ; goto to next pointer
DJNZ    scrsort          ; do a full line

POP     HL                ; fetch lowest, POP is done 2x
LD      A,6
SUB     E                ; A out of screen when not met
JR      Z,scrsort-2      ; do 2 lines

LD      (DE),A          ; out of screen marker for hr

frloop  LD      B,2
LD      A,0              ; turbulence counter
DEC     A
JR      NZ,speeddelay

LD      HL,turbolf       ; show TURBULENCE text
LD      (HL),A

CALL    rnd
AND     7
LD      (turbsh+1),A     ; set extra screenlines

speeddelay LD      HL,1001      ; 1001 on start
spdell   DEC     HL
LD      A,H
OR      L
JR      NZ,spdell
JP      lptst

rnd      PUSH    HL
PUSH    DE
LD      HL,(frames)
rndseed  LD      DE,0
ADD     HL,DE
INC     HL
LD      A,H
AND     #1F
LD      H,A
LD      (rndseed+1),HL
LD      A,(HL)
POP     DE
POP     HL
RET

digitsc LD      (HL),28
DEC     HL
addscore INC     (HL)
LD      A,(HL)
CP      38
JR      Z,digitsc
LD      A,score*256/256+3
CP      L
JR      C,endadd        ; each hundred points speed up

speedup LD      HL,(speeddelay+1)
PUSH    BC
LD      BC,10
SBC     HL,BC

```

```

endadd      POP  BC
            JR   C,endadd
            LD   (speeddelay+1),HL
            XOR  A
            RET

hr          LD   B,14
hr0         DJNZ hr0

            LD   HL,score+#8000      ; low res lines
            LD   A,#1E
            LD   I,A
            LD   A,L
            LD   A,#F5
            LD   BC,#208
            CALL #2B5

hr01        LD   B,9
            DJNZ hr01

turbsh      LD   HL,#4000            ; indexed datatable
            LD   DE,#43C1
            LD   A,0                ; on TURBULENCE screenlines are added
            ADD  A,184
            LD   B,A                ; number of screenlines
            LD   A,D
            LD   I,A                ; set high of screen
hrloop      LD   A,(HL)              ; a line to show?
            CP   B
            JR   NZ,testiron        ; show iron?
            INC  HL                  ; line to show
            LD   A,(HL)              ; fetch dataindex
            INC  HL                  ; point next line
            JP   lbuf+#8000          ; do the display

lbuf        LD   R,A                ; set low of screen
nodisp      DEC  E                  ; in lowmem undo increase
            DEFW 0,0,0,0,0,0,0,0
            DEFW 0,0,0,0,0,0,0,0

jplowbuf    JP   lowbuf             ; 48K bug

lowbuf      NOP                    ; delay
            DJNZ hrloop

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

testiron    LD   A,(DE)              ; pointer of iron
            CP   B                  ; Show iron?
            INC  DE                  ; first step
            NOP                     ; delay
            JR   NZ,nodisp           ; not met, show blank line
            LD   A,E
            INC  E                  ; point to next char
            EX   (SP),HL             ; delay
            EX   (SP),HL
            JP   irondisp+#8000      ; display iron

irondisp    LD   R,A
            NOP                     ; display in middle
            EX   (SP),HL             ; delay
            EX   (SP),HL

```



```

        BIT 0,(HL)                ; 12 tstates highmem 2 bytes
        JP  jplowbuf              ; 48K bug

n      EQU 27
score  DEFB 28,28,28,28,28,28,0
dfile  EQU score
lifecnt DEFB 28

        DEFB 0,0,16,40,17,0,62,55,56,0,0
hisc   DEFB 28,28,30,28,29,35,118

; the turbulence textline
turbolf DEFB 118,0,0,0,0,0,0,0
        DEFB "T"-n,"U"-n,"R"-n,"B"-n,"U"-n,"L"-n,"E"-n,"N"-n
        DEFB "C"-n,"E"-n,118

space  EQU #4300-$

        DEFS space                ; filler to screen, free codeable memory

screen EQU $

basic  DEFB 0,201                ; again, screen used as BASIC-start
        DEFW 0
        DEFB 249,212,28
        DEFB 126
        DEFB 143,0,18,0

; the COPY of the program to make it work on 48K ZX81
init   LD HL,#4009              ; where MC starts
        LD SP,#4400
        LD IX,hr
        LD DE,#C009
        LD BC,#400
        LDIR

        LD HL,udgiron           ; UDG iron over sysvar
        LD DE,#4014
        LD C,5
        LDIR

        LD DE,wrinkles           ; UDG wrinkles over sysvar
        LD C,4
        LDIR

        JP begin                ; start game

udgiron DEFB %01111110
        DEFB %01000000
        DEFB %01111110
        DEFB %01000001
        DEFB %01111111

wrinkl1 DEFB %00011000
        DEFB 0

wrinkl2 DEFB %00100010
        DEFB %00011100

vars    DEFB 128
last    EQU $

```