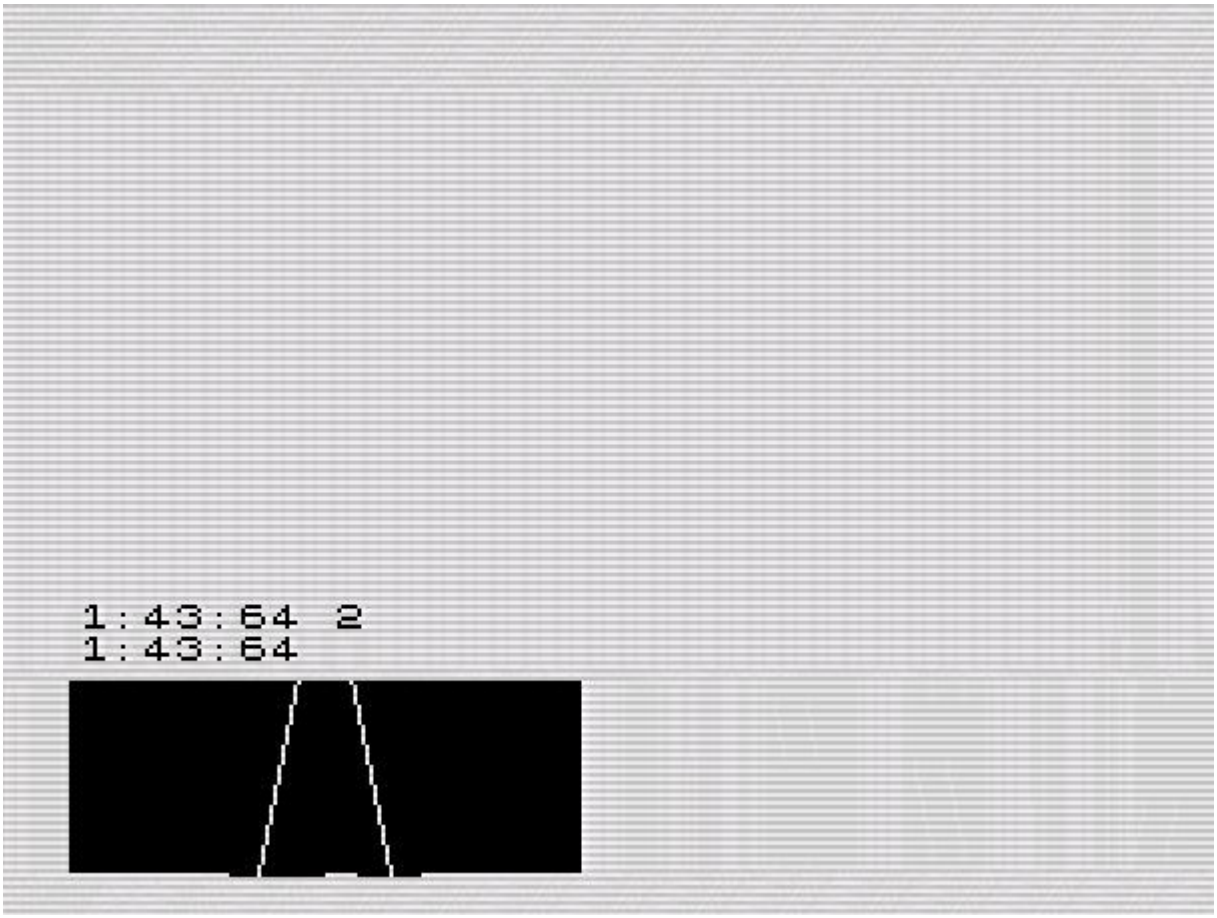
ROADRACER



When you start a new game you always think of a way to compress the screen without losing functionality. A 3D view just has this possibility. Each line only has a few pixels. Can you store just the pixels? The base is set for this game. Each new line must erase the previous line. This is reached by setting a lead position, followed by 2 bytes data. The data is always in the middle so any next byte is always erased. Two lines are drawn on the 'indexed' screen. The indexed screen is translated during the actual display. The nose of the car uses the linedisplay, but due to setting a few checkbits the display is only partly inversed. The hires display is inversed to show the horizon and the air. Even with the display ready it was tight to code a playable game. And yet.... 1K is all you need for a good game.

```
; ROADRACE, a 3D view timeracer in 1k for the ZX81

? * TORNADO *

lstart     EQU  42
rstart     EQU  lstart+33
cctrack    EQU  #4011
nrlaps     EQU  30                ; "2" laps

           ORG  #4009             ;#4009
           DUMP 49161
           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              ; memory above reused for data

berg       DEFB 0
mem        DEFW 0
           DEFB 0
           DEFB 2
           DEFW 1

lastk      DEFB 255,255,255       ; used by ZX81

margin     DEFB 55
nxtlin     DEFW basic
count      DEFB 0
picnr      DEFB 0
flagx      DEFB 0                 ; x
strlen     DEFB 0
tridx      DEFB 0

taddr      DEFW 3213

seed       DEFW 0
frames     DEFW 65535             ; used by ZX81
screenpos  DEFW screen+#800
;coords     DEFB 0,0
prcc       DEFB 188
sposn      DEFB 33,24
cdflag     DEFB 64

cardisp    DEFB 255,255,255,255,255 ; hiresdata of carnose
lwall      DEFB 1,0,0
carstart   DEFB 255,0,128,255,255,255,255,255



hr         LD   B,14              ; sync lowres
           DJNZ hr+2

           LD   BC,#0381
           LD   HL,dfile+#8000
           LD   A,#1E
           LD   I,A
           LD   A,#F5
           CALL #2B5              ; the lowres display

           PUSH HL                ; sync hires
           POP  HL

           LD   HL,#4010          ; erase any plot on the line
hr11       DEC  L
           LD   (HL),0
           JR   NZ,hr11           ; line is clean

hr1        LD   IX,display        ; needed for timing
           LD   BC,#30FF          ; 48 lines visible
           LD   HL,(screenpos)
           LD   D,#40
           LD   A,D
           LD   I,A               ; hiresdata on #4000
```

```
            CALL display2          ; display the tracks

            EX   (SP),HL           ; delay for right display
            EX   (SP),HL           ; nose of car
            PUSH HL
            INC  SP
            INC  SP

            LD   HL,cardisp
            LD   A,L
            LD   IX,exit
            JR   stcar             ; display nose

exit        CALL #292
            CALL #220
            LD   IX,hr             ; set hr start
            JP   #2A4

display     DEC  B
            RET  Z
display2    LD   E,(HL)            ; fetch index
            INC  L
            LDI                    ; set data
            LDI                    ; set data
            LD   E,(HL)            ; fetch index
            INC  HL
            LDI
            LDI                    ; set data
                                   ; set data
            DEC  HL
            INC  HL                ; timing

            LD   A,0               ; for timing not XOR A
stcar       JP   showbuf+#8000     ; goto highmem

showbuf     LD   R,A

track       EQU  $

; the compressed track defines the track.
; each 2 bits is unpacked 8 moves in the game
; visible track is built every 8 moves
; since display has depth 5 in view
; the visible track must be min 7 moves + 5 dept = 12
; the compressed track is hidden in the hires linebuffer
; After loading it is copied with a routine on screen to
; reusable sysvar #4011 to #401E (13 bytes)
; The visible track will be set in the linebuffer.
; Due to the values it can be used together with linebuffer.
; Other tracks must fit 12,5 bytes; 1 nibble needed for double
; start/finish indicator

; l =   0 = turn left
; s =   1 = straight
; r =   2 = turn right
; f =   3 = end of track, start/finish


ctrack      DEFB %01010110         ; sssr
            DEFB %01010001         ; ssls
            DEFB %10101010         ; rrrr
            DEFB %01100101         ; srss
            DEFB %00010000         ; lsll
            DEFB %00000101         ; llss
```

```
            DEFB %10100110          ; rrsr
            DEFB %10010101          ; rsss
            DEFB %01011010          ; ssrr
            DEFB %01011000          ; ssrl
            DEFB %01001010          ; slrr
            DEFB %10010111          ; rssf

            DEFW #8080,#8080        ; ctrack to unused sysvar
                                    ; track here in LBUF
            JP   (IX)               ; back to low


eog         LD   HL,time-1          ; end of game, test hiscore
            LD   DE,hiscore-1
            LD   BC,8
findhi      INC  DE
            INC  HL
            DEC  C
            JR   Z,gamestart
            LD   A,(DE)
            CP   (HL)
            JR   C,gamestart
            JR   Z,findhi
            LDIR


gamestart   LD   A,(lastk)          ; wait for NEWLINE
            SUB  %10111111
            JR   NZ,gamestart

startgame   LD   A,255              ; reset initial startvalues
            LD   (trcnt+1),A
            LD   (tridx),A

            LD   (frames),A         ; reset timecounter

            LD   B,9
            LD   HL,time
            LD   A,28
            LD   (speed+1),A        ; start delay
restimernd  CP   (HL)
            JR   NC,next
            LD   (HL),A             ; reset time and round
next        INC  HL
            DJNZ restimernd

gameloop    LD   A,(laps)           ; finsished ?
            CP   nrlaps
            JR   Z,eog


trcnt       LD   A,0                ; after 8 views built track
            INC  A
            AND  7
            LD   (trcnt+1),A
            JR   NZ,makescreen      ; track is ok for display

            LD   HL,tridx
            INC  (HL)               ; goto next track
            LD   A,(HL)
            LD   HL,cctrack         ; copied ctrack
fstart      LD   E,(HL)
            INC  HL
```

```
            LD   B,A
            SUB  4
            JR   NC,fstart          ; calculate compressed start
            LD   D,(HL)             ; now find within byte
            INC  B
fvaltr      RL   D
            RL   E
            RLA
            RL   D
            RL   E
            RLA
            DJNZ fvaltr
; Bit 0,1 of A holds current track
; Bit 6,7 of E holds next track
            RLC  E
            RLC  E
; now next track in bit 0,1 of E

            LD   HL,track           ; the real decompressed track
            LD   (sor+1),HL         ; also reset start of road
            LD   C,2
built0      LD   B,8                ; repeat track 8x
built1      AND  3                  ; take off remaining fields
            LD   (HL),A             ; set direction on track
            SET  7,(HL)             ; display inverse
            CP   3                  ; is it start/finish?
            JR   C,nosf
            LD   A,(laps)           ; count laps
            INC  A
            LD   (laps),A           ; eog after time add
            XOR  A                  ; full round, back to start
            LD   (tridx),A          ; reset trackdisplay
            INC  A                  ; finish is straight line
nosf        INC  HL                 ; next track to display
            DJNZ built1             ; decompress data
            LD   A,E                ; next track to decompress
            DEC  C
            JR   NZ,built0          ; decompress next track
makescreen  LD   BC,48*6
            LD   D,2                ; do left and right line
            LD   A,lstart           ; start of left line
sroad       LD   HL,(screenpos)     ; screenpos variable for
            ADD  HL,BC              ; hardware with memory
            DEC  D
            LD   (dr5+1),A          ; plotposition left/right
            EXX


sor         LD   HL,track           ; start of road
            LD   B,5                ; 5 pieces of road
            XOR  A                  ; first view without
            LD   (dirvar+1),A       ; previous direction
prway       PUSH HL                 ; save road
            PUSH BC                 ; save depth

            LD   A,(HL)             ; which direction?
            INC  A                  ; #81, #82, #83 or #84
            AND  3                  ;  1, 2, 3 or 0
            LD   (sftest+1),A       ; start finish later
            JR   Z,dirvar           ; sf as straight
            DEC  A                  ;  0, 1, 2
            DEC  A                  ; -1, 0, 1
dirvar      ADD  A,0                ; calculate next view
            LD   (dirvar+1),A
```

```
            ADD  A,A                ; now set direction
            EXX
lrindex     ADD  A,D                ; l=1  r=-1
            EXX                     ; needed to converge lines
            ADD  A,A

            LD   C,A
            LD   E,1
            LD   D,E
            ADD  A,A
            JR   NC,draw            ; dx = positive
            LD   A,C
            NEG
            LD   C,A
            LD   E,255              ; set dx negative

draw        PUSH DE                 ; direction is set, do draw
            LD   B,10               ; dy always 10
            LD   A,C                ; below is shortened DRAW
            LD   L,B                ; from ZX Spectrum
            CP   L                  ; altered to work here
            JR   NC,dr1             ; and off course SHORTENED
            LD   L,C
            XOR  A
            LD   E,A
            DEFB #3A                ; ld a,(nn) skips 2 commands
dr1         LD   B,C
            LD   D,0
            LD   H,B
            LD   A,H
            RRA
dr2         ADD  A,L
            JR   C,dr3
            LD   C,A
dr3         SUB  H
            JR   NC,dr4
            PUSH DE
            EXX
            POP  BC
            JR   dr5
dr4         LD   C,A
            EXX
            POP  BC
            PUSH BC
dr5         LD   A,0                ; plot position
            ADD  A,C
            LD   (dr5+1),A          ; save new plot
            LD   C,A
            XOR  A
            CP   B
            JR   Z,noline

clsline     DEC  HL
            DEC  HL
            DEC  HL
            DEC  HL
            LD   (HL),A
            DEC  HL
            LD   (HL),A
            DEC  HL

            LD   A,C
            AND  #F8
            RRCA
```

```
            RRCA
            RRCA
            LD   (HL),A


noline      LD   A,(HL)
            ADD  A,A
            ADD  A,A
            ADD  A,A
            SUB  C
            CPL
            ADD  A,6
            LD   BC,#8000
doshift     SRL  B
            RR   C
            DEC  A
            JR   NZ,doshift
            INC  HL
sftest      LD   A,0
            OR   A
            LD   A,B
            LD   B,0
            JR   NZ,plot
setsf       LD   B,170
plot        OR   (HL)
            OR   B                  ; insert start/finish
            LD   (HL),A
            INC  HL
            LD   A,(HL)
            OR   C
            OR   B
            LD   (HL),A
            DEC  HL
            DEC  HL

            EXX

            LD   A,C
            LD   (sftest+1),A       ; show Sf only once
            DJNZ dr2                ; do all 48 lines
            POP  DE
            POP  BC
            POP  HL
            INC  HL
            DEC  B
            JP   NZ,prway

            EXX
            LD   BC,48*6+3          ; start of right line
            LD   A,rstart           ; start of right plot
            DEC  D
            JP   Z,sroad            ; do second line or end
; road is now drawn in compressed screen.
; hires will unpack while creating screen

            LD   DE,lwall
fcar        INC  DE                 ; find the car
            LD   A,(DE)
            INC  A
            JR   NZ,fcar
            LD   (DE),A             ; erase car
            LD   C,E                ; save old position

; test track turn drifting
```

```
            LD   HL,(sor+1)
            LD   A,(HL)             ; #80,#81,#82,#83
            XOR  3                  ; #83,#82,#81,#80
            SUB  129                ; 2,1,0,-1
            JR   C,nodrift          ; 2,1,0 finish skipped
            DEC  A                  ; 1,0,-1
            ADD  A,E                ; do the drift
            LD   E,A                ; store back

nodrift     LD   A,(lastk)
            LD   HL,speed+1
            BIT  1,A                ; A-G ; Brake
            JR   NZ,tinc
            INC  (HL)
            JR   Z,incsp1           ; keep some delay
            INC  (HL)
decsp1      INC  (HL)
            DEFB #CA                ; JP Z, never true, hides BIT

tinc        BIT  2,A                ; Q-T ; increase speed
            JR   NZ,tleft
            DEC  (HL)
            JR   Z,decsp1           ; some delay must be kept
            DEC  (HL)
incsp1      DEC  (HL)
                                    ; speedkeys to do
tleft       BIT  3,A                ; 1-5 ; move left
            JR   NZ,tright
            DEC  DE
tright      BIT  4,A                ; 6-0 ; move right
            JR   NZ,testwall
            INC  DE
testwall    LD   A,(DE)             ; test on border hit
            OR   A
            JR   Z,nohit
hitpenalty  LD   E,C                ; undo move
            LD   A,31               ; set speed to low
            LD   (HL),A             ; stil set to speed+1
            XOR  A                  ;
nohit       DEC  A
            LD   (DE),A             ; set car


; the speed is set with framedelay
timing      LD   HL,frames
            LD   A,(HL)
speed       SUB  1
speeddelay  CP   (HL)
            JR   NZ,speeddelay

            LD   A,255
            SUB  (HL)               ; meassure time past
            LD   (HL),255           ; reset time
            LD   B,A
settime     LD   HL,laps-2
            INC  (HL)               ; 1/100
settime2    INC  (HL)               ; 1/50
            LD   A,L
            LD   C,28+6             ; preload minutetest
            CP   min*256/256
            JR   Z,timetest
            LD   C,28+10            ; otherwise 10 test
timetest    LD   A,(HL)
            CP   C
```

```
           JR   NZ,timer
           LD   (HL),28          ; reset current field
findtime   DEC  HL
           LD   A,(HL)
           CP   28
           JR   C,findtime       ; skip  ":"
           JR   settime2         ; increase next field
timer      DJNZ settime



; now we have driven a small piece, so we show next part of road
trdisp     LD   HL,sor+1         ; start of road
           INC  (HL)

           JP   gameloop         ; back to top


dfile      DEFB 118              ; lowres text
time       DEFB 30,14
min        DEFB 33,33,14,28,28,0
laps       DEFB 33,118

hiscore    DEFB 37,14,37,37,14,37,37,0,0,0
           DEFB 118


space      EQU  #43E7-$-314      ; 314 screen + stack
           DEFS space

screen     EQU  $

; the 'screen' is still compressed and unpacked on display
; it is built of an index followed by 2 values set from index
; with this method a line of 20 bytes is just 6 bytes in size.
; the next line will always erase the previous line.
; without the leading 0 a piece of memory would be set as if
; it was a screenindex. Every third is index, not codeable


basic      DEFB 0,0,0            ; only used to start program
           DEFB 0                ; visible track is set over
           DEFB 249,212          ; the basic-loader
           DEFB 31               ; index, but place unused
           DEFB 126,143          ; data
           DEFB 0,18,0           ; index and data
           DEFB 0                ; index
           DEFB 118,0            ; data

           DEFB 0                ; index

init       LD   H,hr/256
           NOP                   ; index
           LD   L,hr*256/256     ; now HL is set
           NOP                   ; index
           PUSH HL               ; store HR
           NOP
           NOP                   ; index
           LD   B,0
           NOP                   ; index
           LD   C,12             ; bc=12
           NOP                   ; index
           LD   D,cctrack/256
           NOP                   ; index
```

```
            LD   E,cctrack*256/256 ; de=cctrack
            NOP                    ; index
            LD   H,ctrack/256
            NOP                    ; index
            LD   L,ctrack*256/256  ; hl=ctrack
            NOP
            LDIR                   ; copy over unused sysvar

            NOP                    ; index
            LD   H,startgame/256
            NOP                    ; index
            LD   L,startgame*256/256 ; HL is start of game

            NOP                    ; index
            POP  IX                ; HR start is set
            NOP                    ; index
            JP   (HL)              ; start game
            DEFB 0                 ; data

            DEFB 0                 ; index
vars        DEFB 128
?
last        EQU  $
```