

## Highwayman



I was driving on the highway when the idea for this game came to me. Ghostbusters taught me to code 3 characters on a 16 byte line. I wanted 3 cars and a player possible in 1 line. I already wanted to bring the size to 12 bytes on a line. Would that time be enough to add an extra character? Although 4 characters needed it are just 2 different characters. This made it possible to display up to 4 characters on a 12 byte line. I added EU and UK driving to the game since I still get hit when crossin the road in Trashman.

```
; Highwayman
; Game 63 in 1K hires for the ZX81

? * TORNADO *

        ORG    #4009
        DUMP 49161

lamp1    EQU    #4000
lamp2    EQU    #400C

basic    LD     L,lamp3*256/256    ; preset for 48K bug
        JR     init0              ; this game has no 48K bug

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

eline    DEFW last                 ; needed to load
chadd    DEFW last-1
xptr     DEFW 0
stkbot   DEFW last                 ; needed to load
stkend   DEFW last                 ; needed to load
berg     DEFB 0
```

```

mem          DEFW 0
            DEFB 0

init1        LD    B,E
            JR    init          ; init can be anywhere

; 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 #16            ; LD d,64
flagx        DEFB 64             ; clever setting of flags

            LD    E,A            ; intruptcounter reset
            EX    AF,AF'

taddr        DEFW 0              ; used by ZX81,no hurting code
            LD    C,24

frames       DEFW #DD*256+1      ; used by ZX81, clever IX set
coprcc       LD    HL,hr         ; set IX
sposn        JR    init1
cdflag       DEFB 64             ; used by zx81

lamptab      DEFB 256-2,12-2
            DEFB lamp3*256/256-2
            DEFB lamp4*256/256-2

;#4040
lane1        EQU    #4040

; key redefinition is programmed over the screen
; the stack writes during intrupt to .f
; so .f is skipped in the code
; or code is corrupted

init         LD    H,D            ; 40
            LDIR                   ; 41 copy data over sysvar
            LD    SP,#4400        ; 43 set SP in free memory
            LD    HL,score+1      ; 46 point to UDLR
            LD    DE,keytab       ; 49 point to keytable
            JR    lane2           ; 4c
            DEFW 0                ; 4e
lane2        LD    A,(lastk)       ; 50 get keycode
            INC    A              ; 53
            JR    NZ,lane2        ; 54 wait for key up
waitkey      LD    BC,(lastk)     ; 56
            LD    A,C            ; 5a
            INC    A              ; 5b
            JR    Z,waitkey       ; 5c wait for key done
            JR    Z,lane3         ; 5e never true
lane3        PUSH    HL           ; 60
            PUSH    DE           ; 61
            CALL    #7BD         ; 62
            LD    A,(HL)         ; 65 get ascii key
            POP    DE            ; 66
            POP    HL            ; 67
            LD    (DE),A         ; 68 write to table
            INC    DE            ; 69
            XOR    A             ; 6a

```

```

LD      (HL),A          ; 6b erase UDLR
INC     HL              ; 6c
OR      (HL)            ; 6d
LD      B,0             ; 6d hide 6f
lane4   JR      NZ, lane2
JP      start           ; start the game

space   EQU      #4080-$
DEFS    space           ; fill to 64 bytes

line    DEFB     170,255,170,170,255,170
         DEFB     170,255,170,170,255,170

lamp3    DEFB     0,255,0,0,165,0
         DEFB     0,165,0,0,255,0

lamp4    DEFB     0,165,0,0,255,0
         DEFB     0,255,0,0,165,0

lineret  DEFB     #DD
LD      L, lampret*256/256 ; show lamps
JR      linedel

low      DJNZ     cloop           ; do all lines
DEC     C
JR      NZ, bloop           ; do second lane

LD      IX, double
LD      B, 5
NOP
JR      linedel+2

double   LD      B, 7
hr00     DJNZ     hr00
DEC     HL                  ; 6 tstate filler
JR      NC, nextlane        ; Carry from stack is end

lastline LD      IX, savesp       ; the final step on screen
RET      NC                  ; 5 tstate filles
LD      A, line*256/256        ; show line
JP      lbuf+#8000

lampret  DEFB     #DD
LD      L, low*256/256
DEC     DE
LD      C, 2+16              ; 2 lanes to show

bloop    EXX
POP      BC                  ; xpos car1 and xpos car2
POP      DE                  ; e xpos car3, d higbyte cars
LD      HL, udgcar           ; graphic of the car
EXX
POP      DE                  ; get xpos player
LD      HL, udgplay          ; the graphic of the player
LD      B, 8                 ; 8 lines per lane
JR      dispin

; the end of HR
savesp   LD      SP, 0
CALL     #292                ; back from intrupt
CALL     #220
LD      IX, hr
JP      #2A4

```

```

; Unlike previous games is the order of the hiresroutine
; broken in parts to keep certain routines within the first
; 256 bytes of the program for timing and memory saving.

; the start of HR
hr      LD    HL,lowres+#8000    ; the lowres display
        LD    BC,#291          ; minimum lines in this game
        LD    A,#1E            ; needed to prevent scrolling
        LD    I,A
        LD    A,#FB
        CALL #2B5

        LD    BC,#C00          ; must be 10 tstates
fill    DJNZ  fill             ; center hires

        LD    (savesp+1),SP

        LD    A,lane1/256
        LD    I,A

        LD    SP,screendata    ; the displaystack

nextlane LD    IX,lineret
        LD    A,line*256/256
        JP    lbuf+#8000

linedel LD    B,7
100     DJNZ 100
        POP   AF               ; get lampline
        INC   DE               ; 6 tstate filler
        JP    lbuf+#8000      ; show lamps

cloop   EXX
        LD    A,(HL)           ; get udg car
        LD    (DE),A          ; write udg car3
        LD    A,E              ; save xpos car3
        LD    E,B              ; get xpos car1
        LDI   ; copy udg car1
        DEC   HL               ; undo change pointer

        INC   BC               ; undo change xpos car2
        LD    E,C              ; get xpos car2

        LDI   ; copy udg car2,set HL pointer
        INC   BC               ; undo change xpos car2
        LD    E,A              ; set xpos car3 back
        EXX   ; 3 cars set, now set player

dispin  LDD                     ; copy udg player
        INC   E
        LD    A,E
        AND   #F0              ; start of data lbuf per lane
        JP    lbuf+#8000      ; show line

keytab  DEFB 54,38,52,53       ; QAOP

lbuf    LD    R,A              ; ED from lbuf is value of no key pressed
allcarx DEFW 0,0,0,0,0,0       ; 12 fields to show
        JP    (IX)

udgcar  DEFB 60,66,255,255
        DEFB 189,66,0

```

```

endplay      DEFB 0                      ; last byte of player udg
             DEFB 102,60,126,231,126,90
udgplay      DEFB 60

screendata   DEFB 0,lamp1*256/256

carlpos1     DEFB lane1*256/256+15      ; car1
car2pos1     DEFB lane1*256/256+15      ; car2
car3pos1     DEFW lane1+15              ; car3
playpos1     DEFW lane1+15              ; player

             DEFB lane2*256/256+15
             DEFB lane2*256/256+15
             DEFW lane2+15
             DEFW lane2+15

             DEFB 0,lamp2*256/256
             DEFB 0,lamp3*256/256

part2        DEFB lane3*256/256+15      ; car1
             DEFB lane3*256/256+15      ; car2
             DEFW lane3+15              ; car3
             DEFW lane3+15              ; player

             DEFB lane4*256/256+15
             DEFB lane4*256/256+15
             DEFW lane4+15
             DEFW lane4+15

             DEFB 1,lamp4*256/256

x            EQU 101
n            EQU 27

ten          LD    (HL),28
             DEC   HL
addsc        INC   (HL)
             LD    A,(HL)
             CP    38
             JR    Z,ten
             RET

deadtest     LD    HL,euk+1
             LD    A,B
             DEC   A
             XOR   (HL)                  ; test UK or EU is reversed
             INC   A
fdead        LD    HL,allcarx+12

startx       DEC   HL
             DEC   HL
             DEC   HL
             DEC   A
             JR    NZ,startx
             LD    E,3                    ; Y now matching
testhit      LD    A,(HL)                ; get Xpos car
             CP    C                      ; test xpos player
             JR    Z,dead
             INC   HL
             DEC   E
             JR    NZ,testhit
             RET

dead         POP   HL                    ; drop return

```

```

LD HL,score-1
LD DE,hiscore-1
LD BC,6
fihi INC HL
INC DE
DEC C
LD A,(DE)
CP (HL)
JR Z,fihi
CALL C,#19F9

start LD HL,euk+1
LD (HL),3 ; default set EU
LD A,247 ; game over, wait for
IN A,(254)
RRA ; newline

JR NC,startgame ; start EU game
RRA
JR C,start ; no 1 or 2 pressed
LD (HL),0 ; set UK game

startgame LD B,5
LD HL,score
ressc LD (HL),28
INC HL
DJNZ ressc

; repair lamps, when done lanes are cleared too.
LD B,4 ; 4 rows of lamps
replane LD C,4 ; 4 lamps per row
replamp CALL lamprep
DEC C
JR NZ,replamp
DJNZ replane ; back for all lines

LD HL,playpos1 ; start of screendata
LD B,4
cl00 LD A,(HL) ; get player online
OR 15 ; set out of line
LD (HL),A ; clear playerposition
LD A,L
ADD A,6
LD L,A
CP part2*256/256
JR NZ,skipl
LD L,part2*256/256+4
skipl DJNZ cl00 ; if before, do next lanes

LD HL,allcarx
LD B,12
clcarx LD (HL),14 ; all cars out of screen
INC HL
DJNZ clcarx

playpos LD BC,#307 ; startposition player

playloop CALL stackpos-3
OR 15 ; make position out of lane
LD (HL),A ; not in lane anymore

; now move player
PUSH BC

```

```

zxpand      LD      BC,%1110000000000111
            LD      A,#A0
            OUT     (C),A
            LD      HL,keytab-1
            IN      A,(C)
            LD      B,5
zxp2key     INC      HL
            ADD     A,A
            JR      NC,dirfound
            DJNZ    zxp2key

;HL holds LBUF when no ZXPAND used

            LD      BC,(lastk)
            LD      A,C
            INC     A
            CALL    NZ,#7BD          ; on key press calculate ASCII address
dirfound    LD      A,(HL)          ; NO KEY reads LBUF
            POP     BC

            PUSH    BC

            LD      HL,keytab
            CP      (HL)
            JR      Z,up
            INC     HL
            CP      (HL)
            JR      NZ,t3

down        LD      A,B              ; preload for test
            INC     B              ; do "move"
            DEFB    33              ; hide DEC B and LD A,B
up          DEC     B              ; do " move"
            LD      A,B              ; load for test

            AND     1              ; test screenposition
            JR      NZ,testmovejr    ; test on move out of screen

lamptest    LD      A,C              ; are we moving to a lamp?
s3          SUB     3
            JR      NC,s3            ; shift lamp by lamp
            INC     A
testmovejr  JR      NZ,testmove      ; we move beside a lamp
; does the lamp need repair?
            POP     BC              ; undo change
            PUSH    BC
            LD      A,C
            DEC     A
            LD      C,0
cntc        INC     C
            SUB     3
            JR      NC,cntc
            CALL    lamppos
            JR      NZ,lampwork
            CALL    lampprep
            LD      HL,score+3      ; 10 points for the repair
            CALL    addsc
lampwork    JR      nomove

t3          INC     HL
            LD      E,(HL)
            CP      (HL)
            JR      NZ,t4
            DEC     C

```

```

t4      INC  HL
        CP   (HL)
        JR   NZ, testmove
        INC  C
testmove LD  A,C
        DEC  A
        CP   12
        JR   NC, nomove
        LD   A,B
        DEC  A
        CP   4
        JR   NC, nomove

        POP  HL                ; valid move, drop old BC
        OR   A
        SBC  HL,BC
        LD   HL,score+4
        CALL NZ,addsc          ; we moved this is 1 point

        DEFB 62                ; hide POP
nomove   POP  BC                ; ilegal move, get old BC

; after move test new position on changed background

        CALL stackpos-3
        LD   (HL),E            ; show player on new pos
        CALL testinv

        CALL deadtest          ; test player moved into car

breaktime LD  A,0
        DEC  A
        AND  31                ; frequency of break
        LD   (breaktime+1),A
        JR   NZ,movecars       ; no break of lamp

        PUSH BC                ; save position player
        CALL rnd
        LD   B,A                ; b now 1..4
        CALL rnd
        LD   C,A                ; c now 1..4

; we have a start, test if it is broken
; if so we test the next. On the start we go round
; to end until a lamp is found
        LD   D,2
cfnd     CALL lamppos           ; test lamp
        JR   NZ,posfnd         ; not broken then break
cdec     DEC  C
        JR   NZ,cfnd           ; test next lamp
        LD   C,4                ; reset lamp in row
        DJNZ cfnd              ; find next lamp
        LD   B,C                ; reset lanes
        DEC  D
        JR   NZ,cfnd           ; when D 255 all lamps broken

posfnd   CALL lampprep          ; use repair to get registers
        LD   A,255
        LD   (HL),A            ; break the lamp
        LD   (DE),A            ; light1 on road off
        DEC  DE
        LD   (DE),A            ; light2 on road off
        DEC  DE

```



```

LD    (DE),A          ; light3 on road off
POP   BC              ; get original BC

CALL  roadpos         ; get roadposition of player
CALL  testinv         ; test if that light broke

movecars  PUSH BC
LD      HL,allcarx
movefreq  LD      A,0
XOR     H
LD      (movefreq+1),A ; only move every 2th step
JR      Z,newcar

moveloop  LD      A,(HL)
CP      13
JR      NC,movedone    ; car is out of view
LD      A,L
CP      allcarx*256/256+6
INC     (HL)
JR      NC,movedone    ; test move left or right
DEC     (HL)
DEC     (HL)
JR      NZ,movedone    ; when zero set out of view
LD      (HL),15        ; out of view
movedone  INC     HL
LD      A,(HL)
ADD     A,A
JR      NC,movefreq

newcar    LD      A,0
DEC     A
AND     7              ; add a car each 8th step
LD      (newcar+1),A
JR      NZ,showcars

LD      E,12
CALL    rnd+2          ; int(rnd*12)+1
LD      HL,allcarx-1

CP      7
JR      C,rightst
LD      E,1
rightst  ADD     A,L
LD      L,A
LD      A,(HL)
CP      13
JR      C,showcars
LD      (HL),E

showcars  LD      D,4
LD      HL,allcarx
sh1      LD      E,3
sh2      LD      C,(HL)
PUSH    HL
PUSH    DE
LD      A,D
DEC     A
EUK      XOR     3      ; EU=3 UK=0
INC     A
LD      B,A
LD      HL,carlpos1-7
LD      A,L
ADD     A,E
LD      L,A

```

```

CALL stackpos
OR 15
LD (HL),A
LD A,(DE)
INC A
JR Z,sh3
LD (HL),E
sh3 POP DE
POP HL
INC HL
DEC E
JR NZ,sh2
DEC D
JR NZ,sh1

POP BC
CALL deadtest ; test cars moved into player

; game delay to keep it playable
delay LD HL,frames ; delay to show screen
LD A,(HL)
SUB 4
wfr CP (HL)
JR NZ,wfr

JP playloop ; do next moves in game

rnd LD E,3
LD A,(frames)
rseed ADD A,#44
LD D,A
RRCA
RRCA
RRCA
XOR #1F
ADD A,D
SBC A,#FF
LD (rseed+1),A
sube SUB E
JR NC,sube
ADC A,E
RET

testinv LD A,(DE) ; get pos in lane
LD HL,udgplay
XOR (HL)
ADD A,A ; test change of bit 7
SBC A,A
LD D,A
invert LD A,(HL) ; get current graphic
XOR D ; invert when needed
LD (HL),A ; save graphic
LD A,L
DEC L
CP endplay*256/256 ; do player
JR NZ,invert
RET

lamprep CALL lamppos ; break lamp pos and lane
LD (HL),165 ; repair lamp
CALL roadpos
ADD A,C
ADD A,C
DEC A

```

```

        DEC    A
        LD     E,A
        XOR    A
        LD     (DE),A           ; repair roadview
        INC    DE
        LD     (DE),A
        INC    DE
        LD     (DE),A
        RET

; routine to calculate lamp and road to break/repair light
lamppos  LD     HL,lamptab-1
        LD     A,L
        ADD    A,B
        LD     L,A             ; point to right lampline
        LD     A,C
        ADD    A,A
        ADD    A,C
        ADD    A,(HL)
        LD     L,A             ; point to right lamp in lane
        LD     A,(HL)
        INC    A
        RET

; HL holds pointer to car1, car2, car3 or player
; routine calculates right position.
stackpos LD     HL,playpos1-6
        LD     DE,6
        LD     A,B
        CP     3
        JR     C,noadd         ; 2 lamps on stack
        ADD    HL,DE           ; must be skipped
        DEC    HL              ; when lane3 or lane4
        DEC    HL              ; is used
noadd    ADD    HL,DE
        DEC    A
        JR     NZ,noadd

; b = 1 to 4 c = 1 to 15
roadpos  LD     D,lanel/256
        LD     A,B
        ADD    A,A             ; b*2
        ADD    A,A             ; b*4
        ADD    A,A             ; b*8
        ADD    A,A             ; b*16
        ADD    A,C             ; b*16+c
        ADD    A,48-1          ; 16 less 64 and 3 less from C
        LD     E,A
        RET

lowres   DEFB 118
score    DEFB 0,"U"+x,"D"+x,"L"+x,"R"+x,0

        DEFB "H"+x,"I"+x,"G"+x,"H"+x,"W"+x,"A"+x
        DEFB "Y"+x,"M"+x,"A"+x,"N"+x,0

hiscore  DEFB 28,28,28,34,31
        DEFB 118,118,118,118

vars     DEFB 128
?
last     EQU    $

```