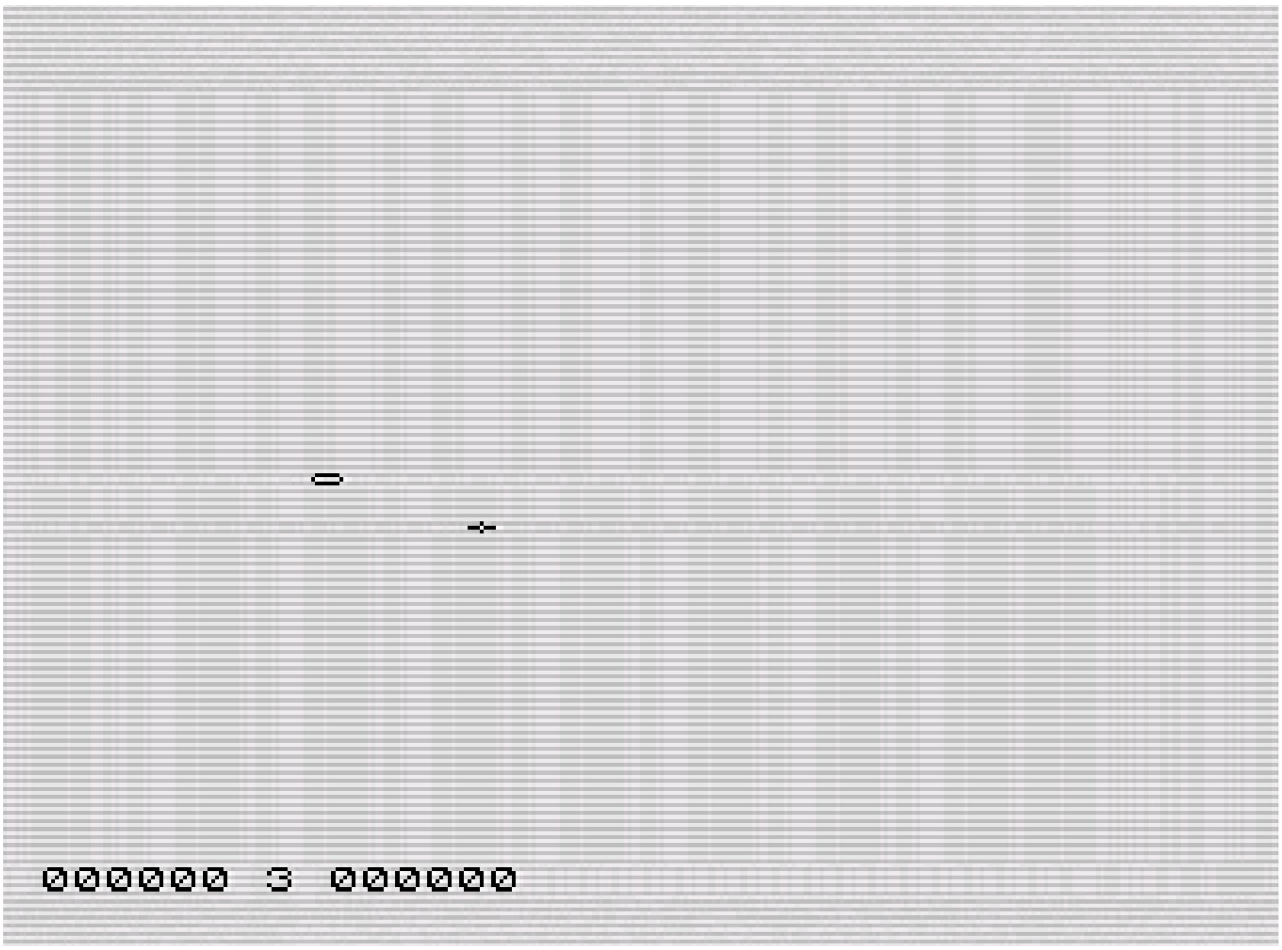
# CLAY PIGEON



**I have used my ONELINERS on the ZX Spectrum more than once to code another hiresgame. During a cleanup I (re)found a CD with a ONELINER for CLAY PIGEON. The screen was easy. Just use the routine from BLOCKY. The gameplay was harder. For a decent gameplay the display had to be as flickerfree as possible and that gave troubles clearing old data. But as usual, I managed so you can enjoy game 28 in 1K hires on the ZX81**

```
; Claypigeon shoot
; 1K hires game for the ZX81 with
; userdefined key controls


? * TORNADO *

           ORG  #4009             ;#4009
           DUMP 49161

           JP   init              ; programmable, fixed address
d_file     DEFW dfile             ; the sysvar needed to load
dfcc       DEFW dfile+1           ; after loading some can be
var        DEFW vars              ; reused for data storage
```

```
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
            DEFB 128
dfsz        DEFB 2
stop        DEFW 1
lastk       DEFB 255,255,255
margin      DEFB 55
nxtlin      DEFW basic
oldppc      DEFW 0
stepcnt     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


marker0     DEFB %00010000,0        ; final value of table here
            DEFB %11101110,0

disctab     DEFB %01111110,%00000000 ; full disc shifted table
            DEFB %10000001,%00000000

            DEFB %00111111,%00000000
            DEFB %01000000,%10000000

            DEFB %00011111,%10000000
            DEFB %00100000,%01000000

            DEFB %00001111,%11000000
            DEFB %00010000,%00100000

            DEFB %00000111,%11100000
            DEFB %00001000,%00010000

            DEFB %00000011,%11110000
            DEFB %00000100,%00001000

            DEFB %00000001,%11111000
            DEFB %00000010,%00000100

            DEFB %00000000,%11111100
            DEFB %00000001,%00000010
```

```
lbuf        LD   R,A                  ; the hires display
sintab      DEFB 3,6,12,18,23,28,33,38 ; table hidden in LBUF
            DEFB 42,46,50,53,55,57,58,59
            DEFB 58,57,55,53,50,46,42
            DEFB 38,33,28,23,18,12,6,3
tabend      JP   (IX)                 ; back to low memory

eog         LD   BC,7                 ; check hiscore
            LD   HL,score-1           ; score index
            LD   DE,hisc-1            ; hiscore index
fihi        DEC  C                    ; full score checked?
            JR   Z,begin              ; only on exact same score
            INC  DE                   ; point to next
            INC  HL                   ; point to next
            LD   A,(DE)               ; check hiscore
            CP   (HL)                 ; against actual score
            JR   Z,fihi               ; still the same, check next
            JR   NC,begin             ; alas, no hiscore, smaller
            LDIR                      ; yeah, set new hiscore


begin       LD   A,%10111111          ; read IN HJKL NL
            IN   A,(254)
            RRCA                      ; Newline to Carry
            JR   C,begin              ; not pressed

            LD   HL,score             ; start game with score reset
            LD   B,6
ressc       LD   (HL),28
            INC  L
            DJNZ ressc

            LD   A,28+5               ; set 5 lives
            LD   (lifecnt),A

            LD   BC,#5550             ; startxy



            EXX
gameloop    JR   Z,eog                ; from above always NZ (INC L)
            LD   HL,#43C0
            XOR  A
cls         DEC  L
            LD   (HL),A
            JR   NZ,cls               ; clear the screenmemory

            LD   DE,sintab            ; start of table
            LD   (current+1),A        ; set start of disc to 0
            LD   (datacp+1),A         ; reset x
            LD   (datacp),A           ; reset y
            LD   B,6
            CALL rnd
            OR   4
```

```
           LD   (rnd2+1),A        ; min width 4
           CALL rnd               ; 1-6
           ADD  A,2               ; 3-8
           LD   (rnd1+1),A        ; set height of claypigeon
           LD   B,100             ; Calculate timer delay
           CALL rnd
           ADD  A,50              ; at least 50 loops delay
           LD   (timer+1),A       ; set release timer
timer      LD   A,0
           DEC  A
           JR   Z,movedisc        ; time has passed
           LD   (timer+1),A       ; store remaining delay
           JR   setmarker         ; do nothing

nextsin    LD   A,(datacp+1)      ; move disc horizontally
rnd2       ADD  A,0
           LD   (datacp+1),A      ; x movement
pos2       LD   HL,0              ; erase old disc
           CALL erudg

           INC  DE
           LD   A,8               ; set minimal timing
           LD   (stepcnt),A

movedisc   LD   A,(DE)            ; fetch sinus value
           LD   B,A
           ADD  A,A
           ADD  A,B               ; x3
           LD   B,A               ; b=SIN*3
           XOR  A
           LD   L,A
           LD   H,3
div8       RR   B
           RRA
           DEC  H
           JR   NZ,div8
           LD   C,A               ; C=remainder of B/8
rnd1       LD   A,8
mula       ADD  HL,BC
           DEC  A
           JR   NZ,mula           ; H = SIN * 3/8 * RND(8)
           LD   A,H               ; next field calculated
           LD   HL,stepcnt
           DEC  (HL)              ; fixed minimal delay
current    CP   0                 ; next sintab reached?
           LD   A,(current+1)     ; fetch current
           JR   NZ,intab          ; not yet reached
           BIT  7,(HL)            ; delay reached?
           JR   NZ,nextsin        ; get next sinusvalue
           DEC  A                 ; undo inc a
intab      INC  A                 ; move to current, either up
           JR   NC,dodisp
           DEC  A                 ; or down
           DEC  A

dodisp     LD   (datacp),A         ; set new Y for display
           LD   (current+1),A      ; and also for calculation


setmarker  EXX
           PUSH BC
           LD   HL,datax+1         ; set coordinates for display
           LD   (HL),C             ; set coordinates on data
           DEC  HL
           LD   (HL),B
           EX   DE,HL
           LD   A,C
           AND  7
           LD   HL,marker0
           JR   Z,markfnd
           DEC  A
markfnd    CALL setdata            ; define shooter to display

           LD   DE,datacp+1
           LD   A,(DE)             ; fetch x of claypigeon
           DEC  DE                 ; point to y
           AND  7
           ADD  A,#10              ; disctab lowbyte /4
           CALL setdata            ; define disc to display


pos1       LD   HL,0               ; point to old pos of shooter
           LD   A,1
nr         SUB  1
           LD   (nr+1),A           ; for visibility
           CALL Z,erudg            ; erase each 2 loops


makescreen LD   BC,screen          ; the compressed screen
           LD   DE,datax           ; data shootcross
           LD   HL,datacp          ; data claypigeon
doline     LD   A,(DE)             ; which comes first,
           CP   (HL)               ; cross or claypigeon?
           CALL C,setline          ; set data of claypigeon
           EX   DE,HL              ; swap registers
           CALL NC,setline         ; set data of cross otherwise
           EX   DE,HL              ; undo swap
           JR   NZ,lineset         ; not on same line, ready now
           LD   A,10               ; LD A,(BC)
           CALL mergeline          ; merge claypigeon same line

lineset    LD   A,32               ; the line is set, point to
           ADD  A,C                ; next free linespace
           LD   C,A                ; point to next line
           XOR  A                  ; mark next line unused
           LD   (BC),A
           LD   A,(DE)             ; test if both UDG's are set
           OR   (HL)
           JR   NZ,doline

           POP  BC                 ; retrieve XY
           PUSH BC                 ; save for false move

           LD   HL,keys            ; start of defined keys
           CALL readkb             ; read first key, UP
           JR   Z,noup
           INC  B                  ; up
noup       CALL readkb             ; read down key
           JR   Z,nodown
           DEC  B                  ; down
nodown     CALL readkb             ; read left key
           JR   Z,noleft
           DEC  C                  ; left
noleft     CALL readkb             ; read right key
           JR   Z,hit
           INC  C                  ; right

hit        LD   A,(datacp)         ; now test for hit
           LD   HL,datax
           SUB  (HL)
           ADD  A,2
           CP   5
           JR   NC,testmove        ; out of y-range
tst2       LD   A,(datacp+1)
           INC  HL
           SUB  (HL)
           ADD  A,8
           CP   15
testmove   POP  HL                 ; drop xy
           JR   C,handlehit        ; in x-range and y-range = hit

           LD   A,B
           SUB  3
           CP   190                ; out of screen top/bottom
           JR   NC,falsemove
           LD   A,C
           SUB  20                 ; out of screen left/right
           CP   213
           JR   C,okmove
falsemove  LD   B,H                ; undo move
           LD   C,L
okmove     EXX
           LD   A,(DE)
           CP   221                ; end of sinus reached?
           JP   NZ,timer
           LD   HL,lifecnt         ; if so, not shot, 1 life down
           DEC  (HL)
           LD   A,(HL)
reljp      CP   28                 ; test for end of game
```

```
           JP   gameloop

handlehit  EXX
           LD   A,(datacp)        ; fetch y-coordinate
           LD   B,A               ; this is your score
addscore   LD   HL,score+6
           DEFB #3A               ; LD A,(NN), hide next command
tens       LD   (HL),28
           DEC  HL                ; point to correct score byte
           INC  (HL)              ; up 1 point
           LD   A,(HL)            ; test on CARRY
           CP   38
           JR   Z,tens
           DJNZ addscore          ; do full score
           JR   reljp             ; saves a byte

; erase udg, either disc or cross
erudg      LD   B,6               ; 6 possible screenlines
           LD   A,L
           AND  31
           LD   L,A               ; point to start of screen
er1        XOR  A
           LD   (HL),A            ; clear field
           INC  HL
           LD   (HL),A            ; and adjusting field
           LD   A,L
           ADD  A,31              ; calculate next line
           LD   L,A
           DJNZ er1               ; do 6 lines
           RET

rnd        PUSH DE
rndsd      LD   DE,0              ; seed
           LD   HL,(frames)       ; timer added too
           ADD  HL,DE
           INC  HL                ; next random field
           LD   A,H
           AND  #1F               ; keep in ROM
           LD   H,A
           LD   (rndsd+1),HL      ; new seed
           LD   A,(HL)
frnd       SUB  B                 ; compute in range needed
           JR   NC,frnd
           ADC  A,B
           POP  DE
           RET                    ; A holds RND upto value in B

readkb     LD   A,(HL)            ; get inport
           INC  HL
           IN   A,(254)           ; do read
           CPL                    ; invert result
           AND  (HL)              ; test against bit of key
           INC  HL                ; point to next key
```

```
          RET

; keys are set, but will be altered by INIT-routine
keys      DEFB %11111011         ; IN Q-T
          DEFB %00000001         ; Q

          DEFB %11111101         ; IN A-G
          DEFB %00000001         ; A

          DEFB %11011111         ; IN Y-P
          DEFB %00000001         ; P

          DEFB %11011111         ; IN Y-P
          DEFB %00000010         ; O

setdata   ADD  A,A               ; calculate correct table valu
          ADD  A,A
          LD   L,A
          PUSH HL                ; copy coordinates and graphic
          PUSH DE
          LD   BC,#204           ; 2x line copy C for later use

sdloop    LD   H,D               ; swap lines
          LD   L,E
          INC  DE                ; make DE point to next line
          INC  DE
          INC  DE
          INC  DE
          LD   A,(HL)            ; fetch y
          DEC  A                 ; 1 line lower next display
          LD   (DE),A            ; save y
          INC  HL
          INC  DE
          LDD                    ; copy x and undo INC DE
          DJNZ sdloop
          POP  DE                ; fetch cross or disc
          POP  HL                ; fetch datapointer
          INC  DE
          INC  DE                ; point to data-address

          LD   B,4               ; LDI will do 1 extra DEC B
setloop   LDI                    ; copy correct data to display
          LDI                    ; 2 bytes always
          LD   A,L               ; table read in correct order
          ADD  A,C               ; 3x C is set by LDI
          ADD  A,C
          LD   L,A
          INC  DE
          INC  DE
          DJNZ setloop
          RET                    ; correct UDG copied

datax     DEFB 40,70,%00010000,0
```

```
           DEFB 39,70,%11101110,0
           DEFB 38,70,%00,0
           DEFB 0                  ; endmarker
datacp     DEFB #03,87,%01111110,0
           DEFB #02,80,%10000001,0
           DEFB #01,80,%01111110,0
           DEFB 0                  ; endmarker

setline    LD   A,175              ; value to set data "XOR A"
mergeline  PUSH BC
           PUSH AF                 ; save both for return
           LD   (setadd1),A        ; "XOR A" or "LD A,(BC)"
           LD   A,(HL)             ; fetch linenr
           LD   (BC),A             ; set on screen
           INC  HL                 ; now to data
           PUSH HL
           LD   H,(HL)             ; fetch x
setspace   INC  BC
           LD   A,H
           SUB  8
           LD   H,A
           JR   NC,setspace        ; go to start of X/8
           XOR  A
           LD   (BC),A             ; set a leading space
           INC  BC
           POP  HL
           INC  HL
           POP  AF
           PUSH AF
           CALL setadd             ; set first part of UDG
           CALL setadd1            ; set second part of UDG

nohit2     POP  AF                 ; fetch old flags
exit       POP  BC                 ; fetch old line
           RET


setadd     JR   NC,setadd0
           LD   (pos2+1),BC        ; claypigeon data
           JR   setadd1
setadd0    LD   (pos1+1),BC        ; cross data
setadd1    XOR  A                  ; CLEAR or fetch screenvalue
           OR   (HL)               ; merge with setvalue
           LD   (BC),A             ; set value to screen
           INC  HL                 ; point to next
           INC  BC                 ; point to next
           RET                     ; return

hr         LD   B,6                ; sync screen
hr0        DJNZ hr0
           LD   HL,screen          ; hires screen pointer
           LD   DE,31              ; line length
           LD   IX,hr2             ; return from high memory
           LD   B,192            ; 192 lines on screen
hr1        LD   A,H
           LD   I,A              ; set I-register
           LD   A,(HL)           ; fetch next linenumber
           INC  L
           CP   B                ; compare with current
           LD   A,L
           JP   Z,lbuf+#8000     ; on equal, display screen
           DEC  HL               ; otherwise undo increase
           LD   C,8              ; and do delay for time needed
nohi       DEC  C                ; do display nothing
           JR   NZ,nohi

           LD   A,I              ; timing
           DEFB 62               ; hide addition
hr2        ADD  HL,DE            ; from display calc new line
           DJNZ hr1              ; do all lines

           EX   (SP),HL          ; sync delay
           EX   (SP),HL
           PUSH HL
           POP  HL

           LD   HL,score+#8000   ; the lowres screen
           LD   A,#1E            ; the index of ROM-characters
           LD   I,A
           LD   A,L
           LD   A,#F5
           LD   BC,#108          ; 1 visible lowresline
           CALL #2B5             ; show lowres

           CALL #292             ; prepare for return to code
           CALL #220
           LD   IX,hr            ; set hires mode again
           JP   #2A4             ; exit through ROM

score      DEFB 28,28,28,28,28,28,0 ; "000000 "
dfile      EQU  score
lifecnt    DEFB "U"-27,0         ; used for define keys
hisc       DEFB 28,28,28,28,28,28 ; "000000"
           DEFB 118

space      EQU  #4300-$          ; screen starts on #4300

           DEFS space

screen     EQU  $                ; screen 1x used for init

basic      DEFB 0,201            ; linenr
           DEFW 0                ; linelength, irrelevant nr
           DEFB 249,212,28       ; RANDOMIZE USR 0
           DEFB 126              ; marker of FP number
           DEFB 143,0,18,0,0     ; #4009 in FP notation
           DEFB 118                ; end of line

; the COPY of the program to make it work on 48K ZX81
init       LD   SP,#4400           ; SP to end of RAM
           LD   IX,hr              ; hires mode activated
           LD   HL,#4000
           LD   DE,#C000
           LD   BC,#400
           LDIR                    ; 48k bug repair

           LD   HL,markertab       ; now set markertable over
           LD   DE,#4000           ; useable sysvar
           LD   C,28
           LDIR

           LD   HL,keys            ; keycontroltable
           LD   DE,dirs            ; directions on screen
           LD   B,4                ; 4 directions to define

upkey      LD   A,(lastk)          ; wait for no keypress
           INC  A
           JR   NZ,upkey

downkey    LD   A,(lastk)          ; wait for keypress
           INC  A                  ; this works with intrupts
           JR   Z,downkey


w4k        LD   C,#FE              ; now check which key pressed
nokeypr    RLC  C
           LD   A,C
           IN   A,(254)
           CPL
           AND  31
           JR   Z,nokeypr          ; check next port
           LD   (HL),C             ; key found, C holds IN-port
           INC  HL
           LD   (HL),A             ; A holds single bit of key
           INC  HL
           LD   A,(DE)             ; fetch next direction
           LD   (lifecnt),A        ; display on screen
           INC  DE                 ; point to next direction
           DJNZ upkey

           JP   begin              ; keys defined, start game

dirs       DEFB "D"-27,"L"-27,"R"-27,28


markertab  DEFB %00001000,0        ; markers are bit shifted
           DEFB %01110111,0

           DEFB %00000100,0
```

```
            DEFB %00111011,%10000000

            DEFB %00000010,0
            DEFB %00011101,%11000000

            DEFB %00000001,0
            DEFB %00001110,%11100000

            DEFB %00000000,%10000000
            DEFB %00000111,%01110000

            DEFB %00000000,%01000000
            DEFB %00000011,%10111000

            DEFB %00000000,%00100000
            DEFB %00000001,%11011100


vars        DEFB 128                ; always last obliged byte
last        EQU  $
```