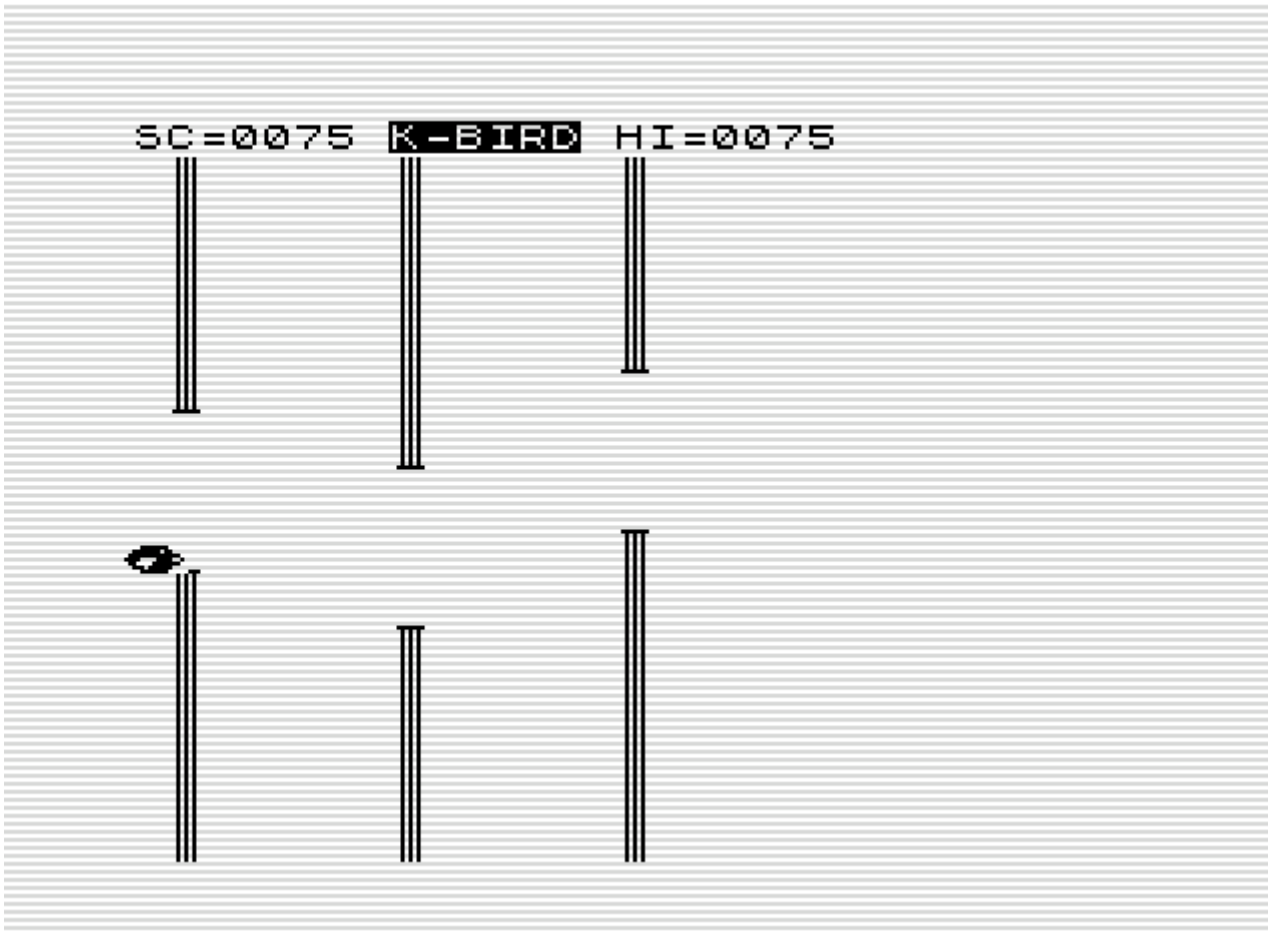


K-Bird



Suddenly I saw a way to code FLAPPY BIRD in 1K hires.
This version is called K-BIRD, since it runs in 1K and you need the keyboard
to flap the wings. The scrolling needed some perfection, this is now reached during intrupt.

```
; K-Bird
; Flappy Bird in 1K hires on the ZX81

scrlines    EQU    176

? * TORNADO *

                ORG    #4009                ;#4009
                DUMP 49161

; the single BASIC-line is fully coded
; over existing systemvariables
; lineNumber and length is used as code

basic        LD      A,0                    ; Line nr and
L400B         JR      init0                  ; start of program

                DEFB 236                      ; the BASIC-command
                DEFB 212,28,126                ; set in reusable sysvar
                DEFB 143,0,18                  ; #4009 in FP notation

;d_file      DEFW 0
;dfcc        DEFW 0
;var         DEFW 0
;dest        DEFW 0
```

```

eline      DEFW last           ; needed on start
chadd      DEFW last-1
xptr       DEFW 0
stkbot     DEFW last
           DEFW last           ; memory above reused for data

berg       DEFB 0
mem        DEFW 0

init0      EX   AF,AF'
           JP   init

lastk      DEFB 255,255,255     ; used by ZX81
margin     DEFB 55             ; used by ZX81

nxtlin     DEFW basic
           DEFB 0
           DEFB 0

flagx      DEFB 0              ; x
strlen     DEFW 0

taddr      DEFW 3213

seed       DEFW 0
frames     DEFW 65535          ; used by ZX81
coords     DEFB 0,0
prcc       DEFB 188
sposn      DEFB 33,24
cdflag     DEFB 64             ; fixed value

hr         LD   A,0             ; HR starts with current
           AND  3               ; dx movement
           LD   H,#40           ; and sets right pointers
           ADD  A,A             ; to make a correct screen
           LD   L,A
           LD   A,(HL)
           LD   (del4to7),A     ; some delay
           INC  L
           LD   A,(HL)
           LD   (del7to4),A     ; sync delay with 207 Tstates

           LD   A,(hr+1)         ; sync byte shift
           RRCA
           RRCA
           AND  7
           LD   E,A
           LD   HL,jump04+#8000
           LD   A,L
           SUB  E
           LD   L,A
           PUSH HL              ; set pixel4 delay
           LD   HL,ret04
           LD   A,L
           ADD  A,E
           LD   L,A
           PUSH HL              ; sync pixel4 in 207 tstates

           LD   HL,dfile+#8000   ; display lowres screen
           LD   BC,#208
           LD   A,#1E
           LD   I,A

```

```

LD    A,#F0
CALL  #2B5

LD    B,4
hr00  DJNZ hr00

hr1   LD    BC,#1313          ; 3 wall starts
p34   LD    E,#13

      POP    IX                ; get return
      POP    HL                ; get jump

      EXX
      LD    DE,23              ; nr bytes on birdscreen
      LD    B,scrlines

      LD    HL,init+256+13      ; start of birdscreen

      CALL  scrloop            ; display the screen

      CALL  #292                ; and back to program
      CALL  #220
      LD    IX,hr
      JP    #2A4

ret04  NOP
ret08  NOP
ret12  NOP
ret16  NOP
ret20  NOP
ret24  NOP
ret28  NOP
ret32  INC    B
      INC    E
      INC    C
      RET    C                ; delay only, never true

      EXX

del4to7 LD    A,(HL)           ; 6-4-2 delay pointer to sync
      LD    C,H                ; C high enough to prevent 0

scrret DEC    B
      RET    Z

scrloop LD    A,(HL)
      SUB    B
      JR    Z,doscreen

      LD    A,init/256          ; display the walls
      LD    I,A

del7to4 NOP
      EXX
      JP    (HL)

doscreen LD    A,H              ; display the bird
      LD    I,A
      LD    A,L
      ADD    HL,DE
      LD    R,A
      JP    lbuf+#8000

```

```

jump32    LD    B,B
jump28    LD    B,B
jump24    LD    B,B
jump20    LD    B,B
jump16    LD    B,B
jump12    LD    B,B
jump08    LD    B,B
jump04    LD    A,C
          AND    127                ; stay in first 128 bytes
          LD    C,A
          LD    R,A
          NOP                ; wall1
          LD    A,B
          AND    127
          LD    B,A
          LD    R,A
          NOP                ; wall2
          LD    A,E
          AND    127
          LD    E,A
          LD    R,A
          NOP                ; wall3
          JP     (IX)

start      LD    A,(lastk)
          SUB    %11111101        ; read A-G to start
          JR     NZ,start

          LD    HL,init+256+12
          LD    B,180
cls        INC    L                ; clear birdscreen
          LD    (HL),A
          DJNZ   cls
          LD    (loop1+1),A        ; reset wallhit

          LD    A,12
          LD    (delay+1),A        ; reset speedup

          LD    HL,#1C1C          ; "00"
          LD    (hr1+1),HL        ; use also as start walls
          LD    (sc),HL           ; reset score
          LD    (sc+2),HL         ; reset score

          LD    A,105             ; reset start of bird
          LD    (ypos+1),A

loop0      LD    B,28

loop1      LD    A,0              ; test wall hit
          OR     A
          JR     NZ,start        ; gameover

          LD    A,B
          CP     6
          PUSH   BC
          CALL   Z,addsc          ; a wall passed=1 point
          POP    BC

; movement
          LD    A,%000000010      ; all rows but A-G
          IN     A,(254)

```

```

CPL
AND 31

LD HL, ypos+1
LD A, (HL)

JR Z, godown ; no key, go done

CP 191
JR NC, nomove ; don't go out of top
ADD A, 4

godown CP 9
JR C, nomove ; don't go out of bottom
DEC A
DEC A
LD (HL), A ; new ypos

nomove PUSH BC
LD A, B
LD (hr+1), A ; save to calc dx in hr
DEC A
RRCA
RRCA
AND 7
LD C, A

; some pixels are not cleared, do manually
LD HL, init+256+14
LD D, H
clr1st XOR A
LD E, L
INC E
INC E
INC E
LD (HL), A
LD (DE), A ; clear bird
INC L
INC E
LD (HL), A
LD (DE), A ; clear bird
LD A, L
ADD A, 22
LD L, A
SUB 175
JR NZ, clr1st

SUB B
AND 3
LD B, A

LD E, 16
LD A, (hr1+1)
CALL wall2bird ; first wall to birdscreen
LD E, 23
LD A, (hr1+2)
CALL wall2bird ; 2nd wall to birdscreen
LD E, 30
LD A, (p34+1)
CALL wall2bird ; 3rd wall to birdscreen

birdidx LD A, #81 ; point to previous bird
XOR 12 ; point to new bird
LD (birdidx+1), A ; save new as old

```

```

    LD    L,A                ; bird for display selected
seth    LD    H,init/256

ypos    LD    A,100          ; bird y on screen

mkbird  LD    B,7            ; 7 lines of bird
        LD    DE,init+256+13 ; start of birdscreen
        LD    (DE),A         ; set pointer
        INC    DE
        INC    DE            ; bird a bit from border
        INC    DE
        INC    DE
        LD    C,A           ; save pointer
        LD    A,(DE)        ; fetch background
        LDI             ; copy to screen
        EX     DE,HL
        OR     (HL)         ; test with second position
        EX     DE,HL
        LDI             ; copy to screen
        JR     Z,nohit
        LD    (loop1+1),A    ; save hitresult, you're dead
nohit   LD    A,E            ; adjust to next line
        ADD    A,17
        LD    E,A
        LD    A,C           ;
        INC    A            ; DEC A (C is 2x DEC)
        DJNZ   mkbird       ; draw full bird before dead

delay   LD    B,12
cloop   DEC    C
        JR     NZ,cloop
        DJNZ   cloop

        POP    BC           ; fetch original

        DEC    B            ; shift the walls
        JP     NZ,loop1

; make a new wall
        LD    A,(p34+1)     ; e
        LD    HL,(hr1+2)    ; bc
        LD    H,A           ; move wall3 to wall2
        LD    (hr1+1),HL    ; save walls
rnd      LD    DE,0
        LD    HL,(frames)
        ADD    HL,DE
        DEC    HL
        LD    A,H
        AND    #1F
        LD    H,A
        LD    (rnd+1),HL
        LD    A,(HL)
        ADD    A,(HL)        ; make it even
frnd     SUB    36
        JR     NC,frnd
        ADD    A,36
        LD    (p34+1),A     ; set new wall3

        JP     loop0

addsc   LD    B,255         ; preset for speedup
        LD    HL,sc+4
ten      DEC    HL

```

```

        INC    (HL)
        LD     A,(HL)
        SUB    38
        JR     NZ, hitest           ; always C here
        LD     B,A                 ; set for speedup
        LD     (HL),28
        JR     ten

hitest   LD     HL,delay+1
        LD     A,(HL)
        SBC    A,B                 ; speed up each 10 points
        LD     (HL),A
        LD     BC,5
        LD     DE,hi-1             ; test hiscore
        LD     HL,sc-1

hi2      INC    HL
        INC    DE
        DEC    C
        RET    Z
        LD     A,(DE)
        CP     (HL)
        JR     Z,hi2
        RET    NC
        LDIR                     ; adjust hiscore
        RET

space    EQU    #4200-$

DEFS    space

init     LD     IX,hr               ; for display on #4200
        LD     HL,#4000
        LD     DE,#C000
        LD     B,4
        LDIR                     ; 48K bug repair

        LD     HL,tab               ; pixeltable over sysvar
        LD     DE,#4000
        LD     BC,8
        LDIR

        LD     HL,start             ; set up of walls
        PUSH   HL                   ; also on sysvar
        PUSH   DE                   ; and start on stack
        LD     HL,fill
        LD     BC,22
        LDIR
        LD     B,40
        RET                         ; do init and start

fill     LD     HL,init+128
f0        DEC    L
        LD     (HL),%01010100      ; wall graphic
        JR     NZ,f0

;8
        LD     L,87
        LD     (HL),254             ; bottom of wall
f1        INC    L
        LD     (HL),0               ; free space
        DJNZ   f1
        LD     (HL),254             ; top of wall
        RET

;12

```

```

tab      CP      (HL)                ; each 2 make up 11 tstates
        NOP
        INC     BC
        RET     Z
        RET     Z
        DEC     BC
        NOP
        CP      (HL)

        DEFS    #4281-$

birds    DEFB    15,224                ; birds udg
        DEFB    59,176
        DEFB    97,252
        DEFB    240,242
        DEFB    127,252
        DEFB    63,240
bird2    DEFB    15,224
        DEFB    63,176
        DEFB    127,252
        DEFB    240,242
        DEFB    97,252
        DEFB    59,240
        DEFB    15,224

wall2bird LD     D,A                ; wall also on birdscreen
        PUSH    BC
        LD      HL,ypos+1
        LD      A,scrlines
        SUB     (HL)
        ADD     A,D
        LD      L,A
        LD      H,init/256
        LD      D,init/256+1
        LD      A,E
        ADD     A,C
        LD      E,A
        LD      C,7
wll2brd  PUSH    BC
        XOR     A
        LD      C,A
        OR      B
        RES     7,L                ; stay in first 128 bytes
        LD      A,(HL)              ; fetch wall to show
        JR      Z,exitrot
rotloop  RLA                    ; rotate for correct display
        RL      C
        RLA
        RL      C
        DJNZ    rotloop
exitrot  EX      DE,HL
        LD      (HL),B                ; clear old pos
        DEC     HL
        LD      (HL),A                ; set new
        DEC     HL
        LD      (HL),C                ; set new
        LD      C,25
        ADD     HL,BC
        EX      DE,HL
        POP     BC
        INC     HL

```



```

DEC    C
JR     NZ,wll2brd
POP    BC
RET

lbuf    DEFB 0,0
        DEFW 0,0,0,0,0,0,0,0,0,0
        JP   low

low      EXX
        INC  C
        INC  B
        INC  E
        EXX
        JP   scrret

n        EQU 27
x        EQU 101                ; inverse text

dfile    DEFB 118
        DEFB "S"-n,"C"-n,20
sc        DEFB 28,28,28,28,0
        DEFB "K"+x,150,"B"+x,"I"+x,"R"+x,"D"+x,0
        DEFB "H"-n,"I"-n,20
hi        DEFB 28,28,28,28
        DEFB 118

vars     DEFB 128
?
last     EQU  $

```