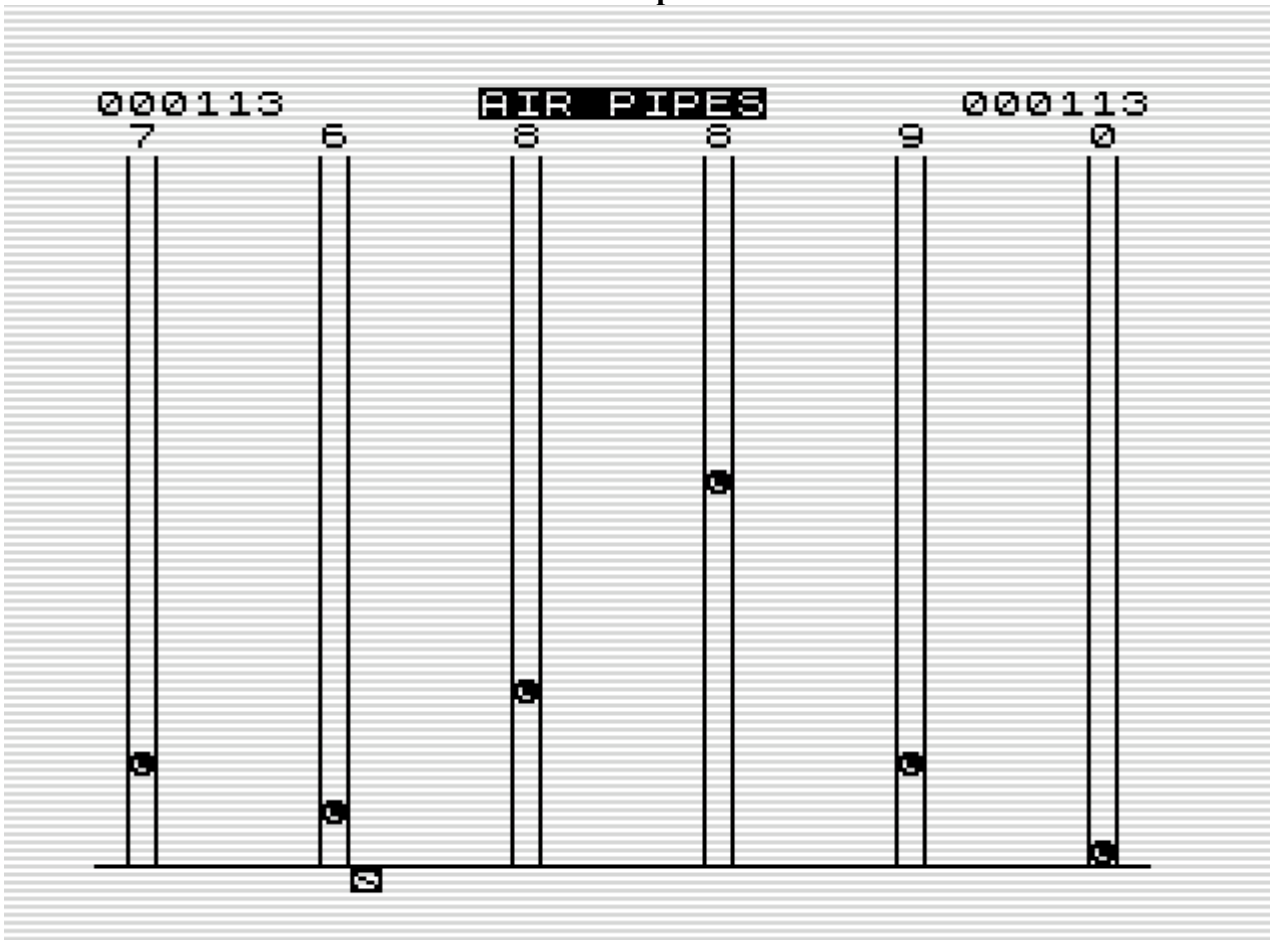


## Air Pipes



**This game was possible, I knew this a long time. Only thing was that the gameplay where it came from was too easy with 6 balls. I had to add some extra tense to make it a playable game. This is done by giving more points for each early captured ball in stead of score after many bounces. Added some speeding up made the game as it is now.**

```
; Airpipes
; Game 34 in 1K hires

? * TORNADO *

scrlines    EQU    176

                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

           ; After loading altered in:
init0      EX   AF,AF'         ; LD R,A
           JP   init           ; DEFB 128
           ; RET

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   HL,dfile+#8000 ; start with lowres screen
           LD   BC,#301         ; A lowres screen with 33 col.
           LD   A,(HL)          ; timing
           NOP                  ; timing
           LD   A,#1E
           LD   I,A
           LD   A,#FD
           CALL #2B5            ; display screen

hr0         LD   B,4             ; timing to start hires ok
           DJNZ hr0
           LD   A,(HL)
           NOP

           LD   A,(udgp+1)      ; swap ventilator
           XOR  5
           LD   (udgp+1),A

ballsy      LD   B,250          ; per register so each 2 steps
           LD   C,250          ; the value can be set
           LD   D,250          ; by the main program
           LD   E,250
           LD   H,250
           LD   L,250
           LD   A,screen/256
           LD   I,A

```

```

        DEFB #DD
        LD    H,scrlines

low      NOP                      ; timing
        NOP

display  INC    B                  ; adjust 6 ballpointers
        INC    C
        INC    D
        INC    E
        INC    H
        INC    L
        DEFB #DD
        DEC    H
        JP     high+#8000         ; display a line

vl       EQU    valveline*256/256

low2     LD     A,#42              ; hires for horizontal line
        LD     I,A
        LD     A,vl-3
        LD     B,6                ; preset display ventilator
        LD     R,A
        LD     A,#40              ; preset hires ventilator
        INC    DE                 ; timing only
        CALL   valves+#8002
        LD     I,A                ; set hires ventilator
xdelay   CALL   p1-0              ; point correct position
udgp     LD     A,0               ; fetch udg ventilator
scr1p    CALL   init0+#8000       ; show ventilator
        CALL   p1-28             ; linefiller
        INC    A                 ; next line of udg
        CP     (HL)              ; timing
        NOP                      ; timing
        DJNZ   scr1p             ; ventilator is 6 lines

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

n        EQU    27
x        EQU    101

dfile    DEFB 118                 ; full lowres screen for init
sc       DEFB 28,28,28,28
tens     DEFB 28,28,0,0,0,0,0,0

        DEFB "A"+x,"I"+x,"R"+x,#80,"P"+x,"I"+x,"P"+x,"E"+x
        DEFB "S"+x
        DEFB 0,0,0,0,0,0
hi       DEFB 28,28,28,28,28,28
        DEFB 118,0
s1       DEFB 28,0,0,0,0,0,0
s2       DEFB 28,0,0,0,0,0,0
s3       DEFB 28,0,0,0,0,0,0
s4       DEFB 28,0,0,0,0,0,0
s5       DEFB 28,0,0,0,0,0,0
s6       DEFB 28
        DEFB 118

valves   LD     R,A

```

```

DEFB #A7          ; inverted with #a7 = AND A
NOP               ; so line is also
DEFW #A7A7,#A7A7 ; used as time filler
DEFB #A7          ; without effect on
NOP              ; value of A
DEFB #A7
DEFW #A7A7,#A7A7
NOP
DEFB #A7
DEFW #A7A7,#A7A7
NOP
DEFB #A7
DEFW #A7A7,#A7A7
NOP
DEFB #A7
DEFW #A7A7,#A7A7
NOP
DEFB #A7
RET

p1

high    LD    A,B
        LD    R,A
        DEFB 0          ; show pipe 1

        LD    A,(HL)
        LD    A,C
        LD    R,A
        DEFB 0          ; show pipe 2

        LD    A,(HL)
        LD    A,D
        LD    R,A
        DEFB 0          ; show pipe 3

        LD    A,(HL)
        LD    A,E
        LD    R,A
        DEFB 0          ; show pipe 4

        LD    A,(HL)
        LD    A,H
        LD    R,A
        DEFB 0          ; show pipe 5

        LD    A,(HL)
        LD    A,L
        LD    R,A
        DEFB 0          ; show pipe 6

        JP    Z,low2    ; 48K bug
        JP    low       ; 48K bug

start   LD    A,(lastk)
        SUB   %11111101 ; wait for A-G to (S)tart
        JR    NZ,start

        LD    (xpos+1),A ; reset startposition

        LD    HL,401     ; reset speeddelay
        LD    (delaycnt+1),HL

        LD    HL,s1      ; scoreindex

```

```

LD BC,6 ; space between scoreindex
EXX
LD B,6 ; 6 valves/scoreindex/scorenrs
; balls/pointers
LD DE,sc ; point to score
LD HL,#4000+11 ; balldata array
LD A,28
ressc LD (DE),A ; reset score
EXX
LD (HL),37 ; reset bouncecount
ADD HL,BC ; point to next
EXX
LD (HL),scrlines-1 ; ball on top
INC HL
LD (HL),scrlines-1 ; max height
INC HL
LD (HL),255 ; set move direction
INC HL
INC DE
DJNZ ressc ; do all 6

gameloop LD A,0 ; each loop movement
DEC A ; is too fast
AND 3 ; therefore 1/4 is a move
LD (gameloop+1),A ; by the player
JR NZ,nomove

LD A,%11101111 ; read 6-0
IN A,(254)
RRA
LD DE,65535
JR NC,xpos ; do right
RRA
JR C,nomove ; no movement key pressed
LD DE,1 ; point to left
xpos LD A,0 ; fetch current position
SUB E
CP 31
JR NC,nomove ; don't go out of range
LD (xpos+1),A ; save move

nomove LD A,(xpos+1) ; fetch current position
LD HL,p1 ; calculate delay
LD E,A ; for right display on screen
XOR A
LD D,A
SBC HL,DE ; delay starts x pos earlier
LD (xdelay+1),HL ; move visible on screen
LD DE,v1 ; point to valve
LD HL,s1 ; point to scoreindex
EXX
LD HL,#4000+11 ; point to ballpointerdata
LD DE,ballsy+1 ; point to ballpointer screen

ballmove LD B,6 ; again 6 balls
LD A,(HL) ; fetch y-pos ball
INC HL ; point to max
CP (HL) ; compare max height
INC HL ; point to direction
JR NZ,noswap ; not top reached
LD (HL),255 ; make it down move
noswap ADD A,(HL) ; do move
DEC HL
DEC HL

```

```

LD      (HL),A          ; save result
ADD     A,80
LD      (DE),A          ; set correct screenpointer
INC     DE
INC     DE              ; point to next ball
CP      85              ; test on bottom reached
LD      A,255           ; preload closed valve
INC     HL
JR      NZ,nobottom     ; no bottom, show closed valve
INC     HL
LD      (HL),1          ; change direction in up
DEC     HL
LD      A,(HL)          ; fetch max height
LD      C,A
RRA          ; A=1/2 A
ADD     A,C             ; A=1 1/2 A
RRA          ; A= 3/4 A
LD      (HL),A          ; height now 3/4 of old
LD      A,6
SUB     B
LD      C,A
ADD     A,A
ADD     A,C
ADD     A,A
LD      C,A             ; C=(6-ballnr)*6 which is XPOS
EXX
LD      A,28
CP      (HL)            ; test value is "0"
JR      Z,scset
DEC     (HL)            ; if not decrease point
scset   EXX
LD      A,(xpos+1)      ; fetch current position
CP      C               ; is it the bottomball?
JR      NZ,newset       ; no, continue
LD      A,scrlines-1
LD      (HL),A          ; set new max heigth
EXX
LD      A,(HL)
SUB     27
LD      B,A             ; calculate reached points
DEC     A
PUSH    HL
PUSH    DE
CALL    NZ,score        ; add remaining points
POP     DE
POP     HL
LD      (HL),37         ; set start to "9"
EXX
newset  LD      A,(HL)   ; fetch current max height
CP      6
JP      C,start        ; ball dropped dead, game over
LD      A,85           ; open valves
nobottom EXX
LD      (DE),A         ; show valves
LD      BC,6
ADD     HL,BC          ; point next scoreindex
EX      DE,HL
ADD     HL,BC          ; point next valve
EX      DE,HL
EXX
INC     HL             ; point next balldata
INC     HL
DJNZ    ballmove

```

```

delaycnt    LD    BC,1000            ; goes down to 1
            LD    HL,399            ; minimum delay 400
            ADD   HL,BC
fr           DEC   HL
            LD    A,H
            OR    L
            JR    NZ,fr
            JP    gameloop

score       LD    HL,sc+6
            DEFB  #3A                ; hide ten-command
ten         LD    (HL),28
            DEC   HL
            INC   (HL)                ; get a point
            LD    A,(HL)
            CP    38
            JR    Z,ten

            LD    A,L
            CP    tens*256/256
            JR    NC,nospeed          ; not over each 100
            LD    HL,(delaycnt+1)    ; current delay
            XOR   A
            LD    DE,50                ; decrease delay =
            SBC   HL,DE                ; increase speed
            JR    C,nospeed          ; not below 0
            LD    (delaycnt+1),HL
nospeed     DJNZ  score                ; do all points won

; hiscore routine is copied from other game
; due to active check score is never the same
            LD    C,7                ; test on hiscore
            LD    HL,sc-1
            LD    DE,hi-1
fihi        INC   HL
            INC   DE
            DEC   C
            RET   Z                    ; same score, impossible
            LD    A,(DE)
            CP    (HL)
            JR    Z,fihi              ; still same score
            RET   NC                  ; lower
            LDIR                ; set new hiscore
            RET

space       EQU   #4300-$-33

            DEFS  space

valveline   DEFB  0
v1          DEFB  255,0,0,0,0
            DEFB  0,255,0,0,0,0
            DEFB  0,255,0,0,0,0
            DEFB  0,255,0,0,0,0
            DEFB  0,255,0,0,0,0
            DEFB  0,255,0

; ball is kept, initlines are overwritten
screen      EQU   $

balludg     DEFB  60+129
            DEFB  126+129
            DEFB  94+129
            DEFB  94+129

```

```

                                DEFB 102+129
                                DEFB 60+129
clr                                DEFB %10000001

vents                            DEFB 0,60,114,78,60
                                DEFB 0,60,78,114,60,0

lbuf                            LD    R,A
                                DEFB 128
                                RET

init                            LD    IX,hr                ; go to HR
                                LD    SP,valveline-1        ; set stackpointer
                                LD    HL,#4000
                                LD    DE,#C000
                                LD    B,3
                                LDIR                        ; repair 48K bug

                                LD    HL,vents              ; copy ventilators
                                LD    DE,#4000
                                LD    C,16
                                LDIR                        ; ventilator UDG on sysvar

                                LD    HL,lbuf               ; copy linebuffer
                                LD    DE,init0
                                LD    C,4
                                LDIR                        ; display vent on sysvar

                                LD    HL,start              ; stack start of game
                                PUSH HL
                                LD    HL,clr
                                LD    DE,clr+1
                                LD    BC,#43FF-clr         ; use ROM LDIR for
                                JP    #19F9                 ; final copy and start

vars                            DEFB 128
last                            EQU    $

```