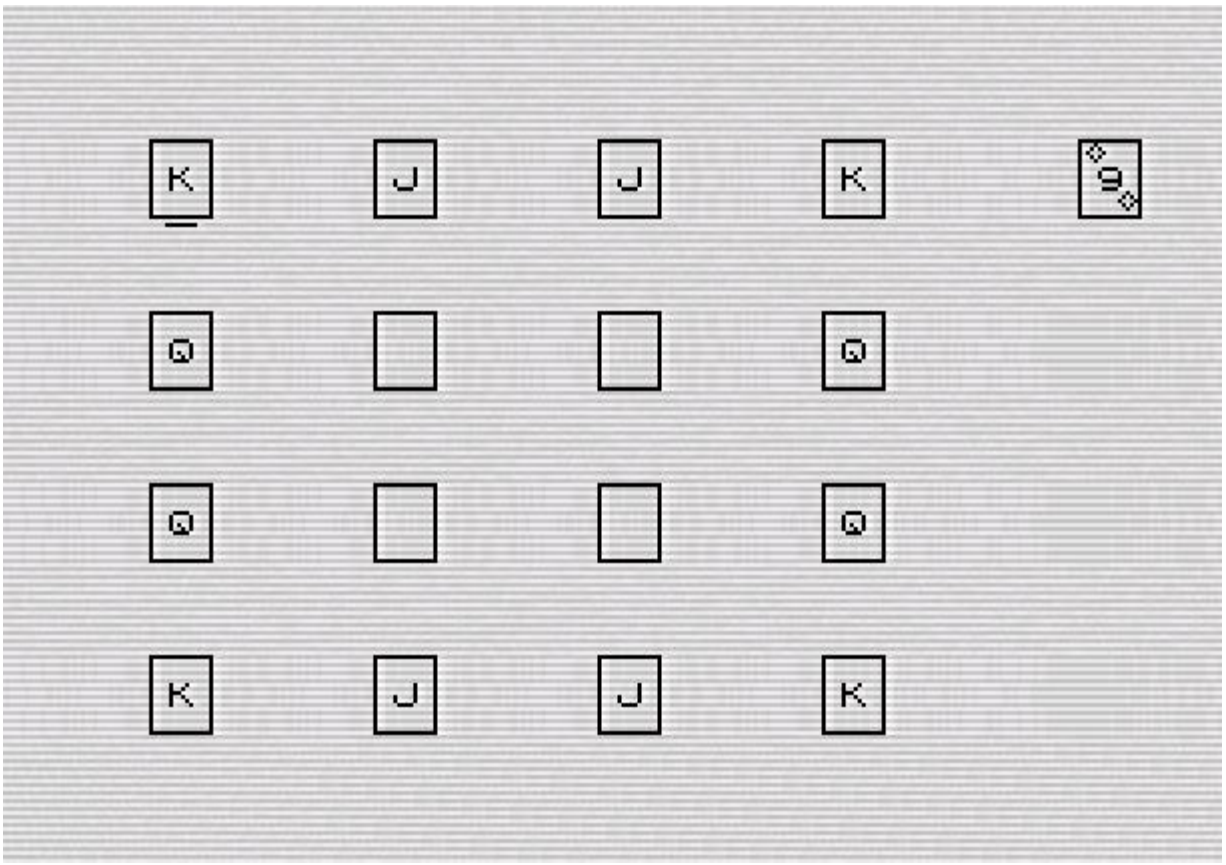
**THE EDGE**



**I coded THE EDGE for the ZX Spectrum. Making a port to the 1K ZX81 was harder than calculated. However we have another 1K hires game ready. The display is in fact 4x the display. In between lines the next screen is built and shown.**
**I had to reduce the building routine a lot. The first version could only display 2 lines. It looks like memory and Spy vs Spy, but those games use fixed graphics where here the screenmap of 5 graphics is shown. Not predefined graphics. The game has all options in it.**

```
; The Edge

; STACK must be on ..00
? * TORNADO *

            ORG  #4009              ;#4009
            DUMP 49161

            JP   init               ; start of program
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                  ; picture counter

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     DEFW 0

taddr      DEFW 3213

seed       DEFW 0
frames     DEFW 65535              ; used by ZX81
card5      DEFW 0
           RET  NZ
;coords    DEFB 0,0
;prcc       DEFB 188
sposn      DEFB 33,24
cdflag     DEFB 64


cpos       DEFS 16                 ; table cursorposition

field      DEFB 1                  ; cards on fields

           DEFB 29*8+1             ; T
           DEFB 17*8               ; H
           DEFB 14*8               ; E
           DEFB 1                  ; _

           DEFB 14*8               ; E
           DEFB 13*8               ; D
           DEFB 16*8               ; G
           DEFB 14*8               ; E

           DEFB 12*8               ; C
           DEFB 1                  ; _
           DEFB 13*8               ; D
           DEFB 27*8               ; R

           DEFB 11*8               ; B
           DEFB 14*8               ; E
           DEFB 14*8               ; E
           DEFB 25*8               ; P

empty      DEFB 255                ; graphical data cards
           DEFB 0,0,0,0,0,0

heart      DEFB 255
           DEFB %00010100
           DEFB %00101010
           DEFB %00100010
           DEFB %00010100
           DEFB %00001000
           DEFB 0

clubs      DEFB 255
```

```
            DEFB %00001000
            DEFB %00011100
            DEFB %00101010
            DEFB %01111111
            DEFB %00101010
            DEFB 0

diamond     DEFB 255
            DEFB %00001000
            DEFB %00010100
            DEFB %00100010
            DEFB %00010100
            DEFB %00001000
            DEFB 0

spades      DEFB 255
            DEFB %00001000
            DEFB %00011100
            DEFB %00111110
            DEFB %00111110
            DEFB %00001000
            DEFB 0
            DEFB 255

; first routines of the program
line2       LD   A,E              ; displaybuffer mid part
            LD   R,A
            DEFB 0,0,0            ; card 1
            ADD  A,44
            LD   R,A
            DEFB 0,0,0            ; card 2
            ADD  A,44
            LD   R,A
            DEFB 0,0,0            ; card 3
            ADD  A,44
            LD   R,A
            DEFB 0,0,0            ; card 4
            SUB  44*4+1
            LD   R,A
            JP   (HL)             ; show card 5 or not

; during init no card is made. Code repaired after init
mkcrdcl     RET                   ; LD   A,(BC) ; fetch card
            EXX
            LD   DE,empty         ; start at empty
            AND  7                ; suit only
            LD   B,A
            ADD  A,A
            ADD  A,B
            ADD  A,A
            ADD  A,E
            ADD  A,B
            LD   E,A              ; point to data suit
            LD   A,(DE)           ; fetch 1st data

setudg      LD   B,7              ; card has 7 top/bottomlines
su1         SET  7,A              ; set left line of card
            LD   (HL),A           ; set top
            INC  L                ; point to next
            LD   C,L              ; save pointer
            LD   A,32
            ADD  A,L
            LD   L,A              ; bottom of card
            INC  E                ; point next of data
```

```
            LD   A,(DE)            ; fetch next
            LD   (HL),A            ; set at bottom
            SET  0,(HL)            ; set right line of card
            LD   L,C               ; retrieve pointer of top
            INC  L                 ; next is top again
            DJNZ su1               ; make suit on card

            EXX
            LD   A,(BC)            ; fetch value again
            INC  C                 ; point to next card
            EXX
            EX   DE,HL
            INC  E
            AND  %11111000         ; value of card only

; since this routine runs on screen all must be sync.
; sync is possible with just 2 bytes

            JR   NZ,vcard          ; 12 /  7
            JR   ecard             ;      12
vcard       ADD  A,224             ;  7

ecard       LD   L,A               ;  4
            LD   A,15              ;  7
            ADC  A,A               ;  4
            LD   H,A               ;  4

            INC  L
            LD   B,6               ; c enough on entry
shloop      LDI                    ; copy from ROM to screen
            INC  E
            INC  E
            DJNZ shloop
            EX   DE,HL
            RET                    ; card ready

cursbuf     LD   R,A               ; the display of the cursor
            DEFB #00
            LD   B,B
            LD   B,B
            ADD  A,1
            LD   R,A
            DEFB #00
            LD   B,B
            LD   B,B
            ADD  A,1
            LD   R,A
            DEFB #00
            LD   B,B
            LD   B,B
            LD   R,A
            LD   L,(HL)            ; trick to increase R
            DEFB #00
            RET  NZ

test11      CP   11                ; 2b stackbuffer now code

card52      DEFB 0,0,0
            RET

stack       DEFS 52                ; 52 positions for the deck

fcardval    AND  #F8               ; erase suit
            LD   E,A               ; save for later
```

```
           LD   HL,#401D          ; cardtable
           LD   C,14              ; cardvalue
findval    CPD                    ; search in table
           JR   NZ,findval
           LD   A,C
           JR   test11            ; saves a RET

hr         EX   (SP),HL           ; the screenroutine
           EX   (SP),HL           ; start on screen
           LD   HL,data           ; the position of card5
           EXX
           LD   BC,field          ; cardpositions, card 5
           CALL mkcrdcl           ; make deck card
           EXX
           LD   E,4               ; 4 loops
           LD   IX,card5+#8000    ; display top/down card5

           LD   HL,sc52+2         ; display mid card5
           SET  7,(HL)

           LD   A,cpos*256/256 -1 ; set cursordata
           LD   (stcur+1),A

hr000      LD   D,4               ; set 4 cards


           EXX
           LD   HL,data+44        ; card1 start

           EXX
mkcrd      CALL mkcrdcl

           LD   A,11
           ADD  A,L
           LD   L,A

cnt2       EXX
           DEC  D
           JR   NZ,mkcrd          ; make 4 cards
           EXX

           LD   B,7               ; now show the cards
           LD   E,data*256/256+44

hrtop      LD   A,data/256        ; show top of card
           RET  C
           LD   I,A
           CALL linebuf+#8000
           INC  E
           INC  E
           DJNZ hrtop

sc52       LD   HL,card52         ; altered for card 5

           LD   B,6               ; show mid of card
hrmid      LD   A,(HL)
           CALL line2 +#8000

l2back     INC  E
           INC  E
           INC  E
           LD   A,(HL)
           DJNZ hrmid
```

```
            LD   A,(HL)
            LD   B,7               ; show bottom of card
hrdown      CALL linebuf+#8000
            INC  E
            INC  E

            PUSH HL
            POP  HL

            DJNZ hrdown

            LD   IX,card5          ; do not show card 5

            LD   B,13
delay       DJNZ delay

            LD   A,#40
            LD   I,A

stcur       LD   A,50
            INC  A
            DEC  HL
            CALL cursbuf+#8000     ; show cursor
            INC  A
            LD   (stcur+1),A

            LD   A,card52/256
            LD   (sc52+2),A

            LD   A,(HL)

            EXX
            DEC  E
            JR   NZ,hr000          ; do all cards

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

linebuf     LD   A,E
            LD   R,A
            DEFW 0
            LD   B,B
            ADD  A,44
            LD   R,A
            DEFW 0
            LD   B,B
            ADD  A,44
            LD   R,A
            DEFW 0
            LD   B,B
            ADD  A,44
            LD   R,A
            DEFW 0
            SUB  44*4+2
            LD   R,A
            JP   (IX)

won         LD   (field),A         ; erase final placed card
            LD   C,170             ; winnercursor
            DEFB 33
begin       LD   C,255             ; lostcursor
            CALL ercurs            ; show current cursors
```

```
gamestart  LD   A,(lastk)         ; wait NEWLINE to start
           SUB  %10111111
           JR   NZ,gamestart
           LD   (count),A         ; set nr free pos to 0

           CALL ercards           ; erase old cards, inc free
;          CALL ercurs-2          ; erase shown cursors

           LD   L,picnr*256/256   ; 12 pictures to set
           LD   (HL),12

           LD   L,seed*256/256
           LD   (HL),52           ; 52 cards in the deck

stop       JR   NZ,playloop       ; double P clears next card

           LD   A,(count)         ; test on back to set cards
           OR   A                 ; while deck is full
           JR   Z,begin           ; result: losing game but also
                                  ; trick to clear startscreen

playgame   LD   HL,frames         ; now pick a random card
           LD   A,R               ; with frames and R
           ADD  A,(HL)            ; A is "random"
           LD   HL,(seed)         ; nr of cards in L
rlp        LD   E,A               ; random card to E
           SUB  L
           JR   NC,rlp            ; repeat until in boundary
           DEC  L                 ; 1 card less
           LD   (seed),HL         ; save remaining number
           LD   D,stack/256
           LD   H,D
; swap (DE) with (HL), move found card out of deck
           LD   A,(DE)            ; the chosen card
           INC  C
           LDI                    ; move out of range in range
           DEC  HL

           LD   (HL),A            ; move selected out of range

           LD   (field),A         ; card to displayfield
           CALL fcardval          ; which value of card
           JR   C,playloop        ; low card always ok
; test if J, Q, K has free position
           LD   HL,#4000+18       ; destined positionstable
testfree   DEC  L
           JR   Z,begin           ; no free places, lost
           LD   A,(HL)
           AND  %11111001         ; take of suit but keep set
           CP   E                 ; card without suit or set
           JR   NZ,testfree       ; card can't be placed

playloop   LD   HL,cpos           ; cursortable
           LD   A,B
           ADD  A,L
           LD   L,A               ; point to cursorplace
           LD   A,(HL)            ; fetch old value
           LD   E,A               ; save old cursor
           CPL                    ; invert cursor
           LD   (HL),A            ; show cursor

           LD   D,L               ; save cursorpos
           LD   L,frames*256/256  ; point to frames
```

```
            LD   A,(HL)
            SUB  6
wfr         CP   (HL)              ; 6 frames delay
            JR   NZ,wfr
            LD   L,D               ; retrieve cursorpos
            LD   (HL),E            ; undo cursor set

            LD   A,(lastk)         ; fetch key
            LD   (errkey),A        ; set errorkey
            LD   HL,dirs-1
fdir        INC  HL
            CP   (HL)
            INC  HL
            JR   NZ,fdir           ; find keypress routine

dirf        LD   L,(HL)            ; HL now jump routine
            LD   A,(field)         ; fetch card
            OR   A                 ; test on zero for later
            LD   A,B               ; fetch cursorpointer
            JP   (HL)              ; goto routine

dirs        DEFB #DF,stop*256/256
            DEFB #FB,up*256/256
            DEFB #FD,down*256/256
            DEFB #F7,left*256/256
            DEFB #EF,right*256/256
            DEFB #7F,hit*256/256
errkey      DEFB 0,playloop*256/256
down        ADD  A,8               ; 8-3-2+1=4
up          SUB  3                 ; -3-2+1=-4
left        DEC  A                 ; -2+1=-1
            DEC  A
right       INC  A                 ; +1
            AND  15                ; stay within boundaries
            LD   B,A               ; new cursorposition
ret2play    JR   playloop

hit2        ADD  A,cpos*256/256    ; erase mode
            LD   L,A               ; HL points to cursorplace
            LD   H,cpos/256

            LD   A,E               ; fetch cursordisplay
            OR   A
            JR   NZ,playloop       ; already selected
            LD   (HL),170          ; set selected mode

            LD   A,17              ; now go to board
            ADD  A,L
            LD   L,A
            LD   A,(HL)            ; get card
            CALL fcardval          ; calculate value
score       ADD  A,0               ; add old score
            CP   10                ; sum 10?
            CALL Z,ercards         ; exact, erase cards, RET=NC
            CALL NC,ercurs-2       ; NC, too high, A becomes 0
continue    LD   (score+1),A       ; set new score
            JR   playloop          ; continue play

hit         JR   Z,hit2            ; test on erase mode
            INC  A
            LD   HL,picnr
            LD   D,(HL)            ; fetch picturenr
            LD   L,A
            BIT  0,(HL)
```

```
            JR   NZ,playloop       ; already set
            LD   A,10
            CP   C                 ; C holds current cardvalue
            LD   A,(field)         ; card on deck
            LD   E,A               ; save card for later
            JR   NC,setok          ; no pict on all fields
            XOR  (HL)              ; check value
            AND  %11111000         ; take of suit/set
            JR   NZ,ret2play       ; false position
            DEC  D                 ; 1 picture placed
setok       SET  0,(HL)            ; position taken
            LD   A,field*256/256+1
            ADD  A,B
            LD   L,A
            LD   (HL),E            ; set card on board

            LD   A,D
            OR   A
            LD   (picnr),A         ; set current open pictures

            JP   Z,won             ; won, all K, Q, J placed

            LD   L,count*256/256
            DEC  (HL)              ; 1 free pos less
            JP   NZ,playgame
; switch gameplay
            XOR  A                 ; board is full
            LD   (field),A         ; erase deck shown
            JR   ret2play

ercards     PUSH BC                ; routine to erase cards
            LD   BC,#4000+16       ; empty place value table
            LD   DE,field+16       ; the cards on field
            LD   HL,cpos+15        ; the cursors
clboard     LD   A,(HL)            ; a cursor here?
            OR   A
            JR   Z,nxtfield

            LD   A,(BC)            ; take of set
            AND  %11111000
            LD   (BC),A            ; unset card
            LD   (DE),A            ; default to board
            LD   A,L               ; save index
            LD   L,count*256/256
            INC  (HL)              ; empty place counter
            LD   L,A               ; restore index
nxtfield    DEC  L
            DEC  E
            DEC  C
            JR   NZ,clboard        ; do entire board
            POP  BC                ; return cursor
;           RET    ; always ercurs after ercard

            LD   C,0               ; c used as cursor
ercurs      LD   A,16              ; routine used to (re)set
            LD   HL,cpos           ; cursor on screen
erc1        LD   (HL),C
            INC  HL
            DEC  A
            JR   NZ,erc1           ; do all fields
            RET

SPACE       EQU  #4322-$           ; some free mem for SP
            DEFS SPACE
```

```
data       EQU  $

; screenmemory is used to define default cardlayout
; where init is replaced by carddata through ROM LDIR
; so initialcode is overwritten and memory double used

init       LD   IX,hr             ; go to 'broken' hr-mode
           LD   SP,data           ; set SP in 1K

; not direct into right hiresmode. In this way I can do the
; the "repair" to use memory to the full

           LD   HL,#4000          ; make a full copy of the
           LD   DE,#C000          ; program above 49152, needed
           LD   BC,1024           ; for the himem commands.
           LDIR                   ; 48K bugrepair

           LD   HL,cards          ; default and values
           LD   DE,#4000          ; over sysvar
           LD   BC,30
           LDIR

           LD   HL,44*4+1+data    ; create fixed corners
           LD   DE,44*4+44+data   ; of cards
           LD   BC,#7FF
           LD   A,C
mk1        LD   (HL),C
           LD   (DE),A
           LD   A,128
           LD   C,1
           INC  L
           INC  L
           DEC  E
           DEC  E                 ; right up and
           DJNZ mk1               ; left down created

           LD   B,6               ; mid part must be set
           LD   HL,4*44+14+data
mk2        LD   (HL),8
           INC  L
           INC  L
           LD   (HL),16
           INC  L
           DJNZ mk2

           LD   DE,stack          ; now create the deck
           LD   B,4
l0         LD   C,13
           LD   HL,#4011          ; from King to Ace
l1         LD   A,(HL)
           ADD  A,B               ; add suit
           LD   (DE),A            ; set card
           INC  DE
           INC  L
           DEC  C
           JR   NZ,l1             ; do all values
           DJNZ l0                ; do all suits

           LD   HL,44*4+data      ; set final card
           LD   DE,data           ; to first card, overwriting
           PUSH DE                ; save dest for next start
           LD   C,44
           LDIR                   ; some piece above
```

```
            LD   HL,begin+2         ; set start on stack
            EX   (SP),HL            ; RET in ROM to start program
                                    ; and start fetched
            LD   C,3*44             ; 2 cards ready, 3 to do

            LD   A,#0A              ; command LD A,(BC)
            LD   (mkcrdcl),A        ; repair the HR
            JP   #19F9              ; through ROM LDIR and RET

basic       DEFB 0,1                ; only used to start program
            DEFW lenbas-$
            DEFB 249,212,28
            DEFB 126
            DEFB 143,0,18,0,0
dfile       EQU  $
lenbas      DEFB 118,0

cards       DEFB 26*8               ;%00100100

            DEFB %10100000          ; K
            DEFB %10011000          ; J
            DEFB %10011000          ; J
            DEFB %10100000          ; K

            DEFB %11010000          ; Q
            DEFB %00000000          ; _
            DEFB %00000000          ; _
            DEFB %11010000          ; Q

            DEFB %11010000          ; Q
            DEFB %00000000          ; _
            DEFB %00000000          ; _
            DEFB %11010000          ; Q

            DEFB %10100000          ; K
            DEFB %10011000          ; J
            DEFB %10011000          ; J
            DEFB %10100000          ; K

cardvalues  DEFB 10*8               ; A
            DEFB 16                 ; 2
            DEFB 24                 ; 3
            DEFB 32                 ; 4
            DEFB 40                 ; 5
            DEFB 48                 ; 6
            DEFB 56                 ; 7
            DEFB 64                 ; 8
            DEFB 72                 ; 9
            DEFB 29*8               ; T
            DEFB 19*8               ; J
            DEFB 26*8               ; Q
            DEFB 20*8               ; K

vars        DEFB 128
?
last        EQU  $
```