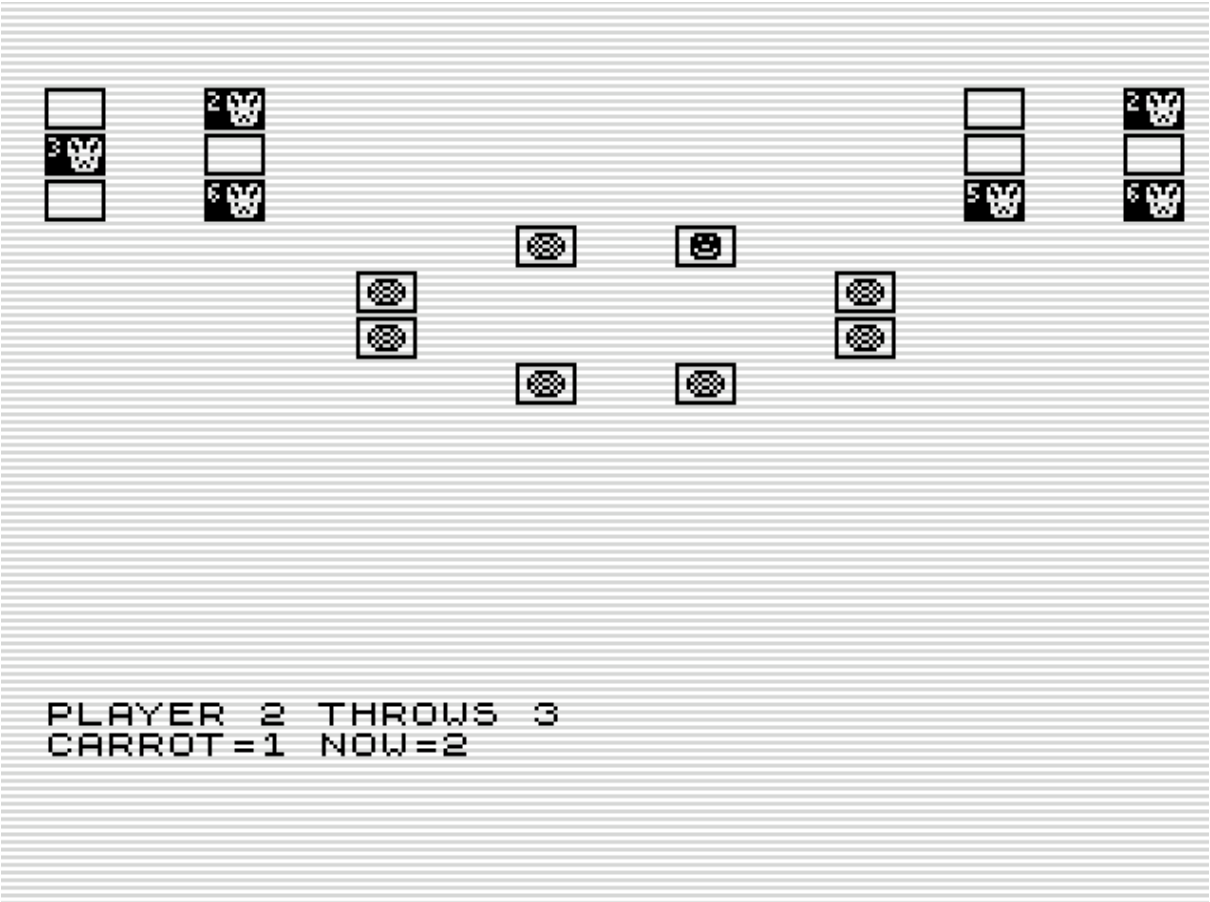
Rabbits



On my vacation I found the boardgame “Max Mümmelman”.
This game is the base for this 1K hires game. The game is 2 player only.
By doing so I was able to add all rules and AI for the computer.
Finally I found the room to repair the 48K-bug.

```
; Rabbits, the game
; Based on "Max Mummelmann", a german boardgame

? * 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   B,3              ; block for 48K copy
L400B      JR   init0            ; do init

           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
init1      LD   E,L                ; E=L for 48K bug
           JP   init               ; final steps init

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

nxtlin     DEFW basic
init0      XOR  A                  ; reset intruptcounter
           DEFB 254                ; hide flagx

flagx      DEFB 0                  ; x
strlen     JR   seed               ; continue

taddr      DEFW 3213

seed       EX   AF,AF'             ; delay intrupt
           DEFB #3A                ; LD A,(NN)
frames     DEFW 65535              ; used by ZX81
coords     JR   init1              ; continue
prcc       DEFB 188
sposn      DEFB 33,24
cdflag     DEFB 64                 ; fixed value

hr         PUSH HL                 ; timing
           INC  HL
           DEC  L
           POP  HL

           LD   A,#41              ; hirespointer
           LD   I,A

           LD   HL,scrindex
           CALL shplay             ; card 1/2 of player 1 and 2
           CALL shplay             ; card 3/4
           CALL shplay             ; card 5/6

           LD   HL,board           ; point to board
           LD   IY,highbrd+#8000   ; displayline can change
           CALL shboard            ; first line
           CALL shboard2           ; second line, other line
           CALL shboard2           ; third line
           LD   IY,highbrd+#8000
           CALL shboard3           ; fourth line

           LD   IY,#4000           ; original IY back

           LD   BC,#354-8
           LD   A,#1E
           LD   I,A
           LD   A,#FB
           LD   HL,dfile+#8000     ; lowres text
           CALL #2B5

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

```
shplay     LD   B,(HL)            ; fetch cardpointer1
           INC  L
           LD   D,(HL)            ; fetch cardpointer2
           INC  L
           LD   C,(HL)            ; fetch cardpointer3
           INC  L
           LD   E,(HL)            ; fetch cardpointer4
           INC  L

           DEFB #DD               ; number of lines in IXh
           LD   H,11

           JR   low2in

lowinc     INC  B
           RET  Z                 ; never true, B=odd here
           INC  B
           DEFB #DD
           DEC  H
           LD   A,D
           EX   (SP),HL
           EX   (SP),HL
           JP   high2+#8000

high2      LD   R,A
           DEFW 0
           LD   A,E
           LD   R,A
           DEFW 0
           JP   NZ,low2           ; test from dec ixh, 48K bug

           EX   (SP),HL           ; line filler
           EX   (SP),HL           ; this is done in "upper"
           EX   (SP),HL           ; memory, so bit 6
           EX   (SP),HL           ; must be set on all opcodes
           EX   (SP),HL
           EX   (SP),HL
           LD   A,(HL)
           LD   A,(HL)
           LD   A,A
           LD   A,A
           RET  Z                 ; always true from above

low2       INC  E                 ; adjust UDG-pointers
           INC  E
           INC  D
           INC  D
           INC  C
           INC  C
low2in     JP   high+#8000

shboard2   LD   IY,highbrd2+#8000

shboard3   NOP
           LD   B,6
del1       DJNZ del1

shboard    LD   B,(HL)            ; fetch cardpointer1
           INC  HL
           LD   C,(HL)            ; fetch cardpointer2
           INC  HL
           LD   A,(HL)
```

```
            LD   D,12

            PUSH HL
            JR   highbrdin

blow2       INC  BC                ; for timing BC
            INC  C
            INC  B                 ; first INC B is done elsewher

            PUSH HL
            EX   (SP),HL
            EX   (SP),HL
            EX   (SP),HL
highbrdin   POP  HL

            LD   A,B
            DEC  D
            RET  Z
            INC  B
            JP   (IY)

highbrd2    LD   R,A
            DEFW 0

            EX   (SP),HL
            EX   (SP),HL
            LD   A,A

            LD   A,C
            LD   R,A
            DEFW 0
            JP   blow2

highbrd     PUSH HL
            POP  HL

            LD   R,A
            DEFW 0
            LD   A,C
            LD   R,A
            DEFW 0

            PUSH HL
            POP  HL

            JP   blow2

r1          EQU  rab1*256/256      ; lowbyte of rab1 only
r2          EQU  rab2*256/256
r3          EQU  rab3*256/256
r4          EQU  rab4*256/256

r5          EQU  rab5*256/256
r6          EQU  rab6*256/256
em          EQU  empty*256/256
ca          EQU  carrot*256/256
cy          EQU  cardyes*256/256
pw          EQU  pawn*256/256

space4100   EQU  #4100-$           ; filler to set UDG on #4100
            DEFS space4100

rab1        DEFB 255,255
            DEFB 222,115
```

```
            DEFB 157,37
            DEFB 221,173
            DEFB 220,137
            DEFB 140,1
            DEFB 254,139
            DEFB 254,35
            DEFB 254,83
            DEFB 255,7
rab2        DEFB 255,255
            DEFB 158,115
            DEFB 237,37
            DEFB 221,173
            DEFB 188,137
            DEFB 140,1
            DEFB 254,139
            DEFB 254,35
            DEFB 254,83
            DEFB 255,7
rab3        DEFB 255,255
            DEFB 158,115
            DEFB 237,37
            DEFB 221,173
            DEFB 236,137
            DEFB 156,1
            DEFB 254,139
            DEFB 254,35
            DEFB 254,83
            DEFB 255,7
rab4        DEFB 255,255
            DEFB 174,115
            DEFB 173,37
            DEFB 141,173
            DEFB 236,137
            DEFB 236,1
            DEFB 254,139
            DEFB 254,35
            DEFB 254,83
            DEFB 255,7
rab5        DEFB 255,255
            DEFB 142,115
            DEFB 189,37
            DEFB 157,173
            DEFB 236,137
            DEFB 156,1
            DEFB 254,139
            DEFB 254,35
            DEFB 254,83
            DEFB 255,7
rab6        DEFB 255,255
            DEFB 206,115
            DEFB 189,37
            DEFB 157,173
            DEFB 172,137
            DEFB 220,1
            DEFB 254,139
            DEFB 254,35
            DEFB 254,83
            DEFB 255,7
empty       DEFB 255,255
            DEFB 128,1
            DEFB 128,1
            DEFB 128,1
            DEFB 128,1
            DEFB 128,1
```

```
            DEFB 128,1
            DEFB 128,1
            DEFB 128,1
            DEFB 128,1
cardyes     DEFB 255,255
            DEFB 128,1
            DEFB 135,225
            DEFB 138,81
            DEFB 149,169
            DEFB 154,89
            DEFB 149,169
            DEFB 138,81
            DEFB 135,225
            DEFB 128,1
pawn        DEFB 255,255
            DEFB 128,1
            DEFB 135,225
            DEFB 141,177
            DEFB 143,241
            DEFB 142,113
            DEFB 139,209
            DEFB 140,49
            DEFB 135,225
            DEFB 128,1
carrot      DEFB 255,255
            DEFB 128,1
            DEFB 128,1
            DEFB 135,245
            DEFB 152,153
            DEFB 162,189
            DEFB 154,25
            DEFB 135,245
            DEFB 128,1
            DEFB 128,1
            DEFB 255,255

high        LD   A,B
            LD   R,A
            DEFW 0
            LD   A,C
            LD   R,A
            DEFW 0
            JP   lowinc             ; 48K bug

            LD   C,pw
card2brd    PUSH HL
            LD   HL,boardtab-1      ; table needed to calc
            LD   A,L                ; displayposition
            ADD  A,B
            LD   L,A                ; point to tableposition
            LD   L,(HL)             ; get displayposition
            LD   (HL),C             ; set on screen
            POP  HL
            RET

rnd         PUSH HL
            LD   HL,(frames)
seedrnd     LD   DE,0
            ADD  HL,DE
            DEC  HL
            LD   A,H
            AND  #1F
            LD   H,A
            LD   (seedrnd+1),HL
```

```
            LD   A,(HL)
frnd        SUB  B
            JR   NC,frnd
            ADC  A,B
            POP  HL
            RET

init        LD   IX,hr              ; goto HR-mode

; initdata in next bytes
; on start used as init and set up cards
; this shows are rubbish on screen
scrindex    EQU  $+1

            LD   D,#BF              ; destination
            LD   H,#3F              ; start
            LDIR                    ; repair 48K bug with copy
            LD   DE,#4000           ; now make the cards
            LD   B,4                ; 4 families
s0          LD   A,5*20             ; of each 6 rabbits
s1          LD   (DE),A             ; index 0,20,40,60,80,100
            INC  DE
            SUB  20
            JR   NC,s1              ; end when below 0
            DJNZ s0

restart     LD   A,(lastk)          ; read A-G for "S"tart
            CP   %11111101
            JR   NZ,restart

; clear cards of players
            LD   HL,scrindex+12
cls         DEC  L
            LD   (HL),em            ; index empty card
            JR   NZ,cls

; shuffle deck of cards and reset cards taken
            LD   HL,#4000+23
shflp       LD   B,L
            CALL rnd
            DEC  A                  ; take 1 less, this will
            LD   D,H                ; then reset #4000 too.
            LD   E,A
            RES  0,(HL)             ; use bit 0 for 'taken', reset
            LD   A,(DE)             ; fetch random position
            RES  0,A                ; erase takenbit
            LDI                     ; swap positions, shuffle
            DEC  L                  ; undo increase
            LD   (HL),A             ; swap complete

            LD   A,29               ; start carrot on 1
            LD   (carpos),A         ; as text visual

            LD   A,L                ; 1 in the end
            DEC  L                  ; point to next card
            JR   NZ,shflp

            LD   L,25
            LD   (HL),A             ; carrot position
            DEC  L
            LD   (HL),A             ; cursor position

            PUSH AF
            CALL mkbrd              ; first cleared screen
```

```
           POP  AF

playloop   LD   (curpl+1),A
           ADD  A,28              ; number to ascii
           LD   (plnr),A

           LD   B,6
           CALL rnd               ; throw a die
           LD   B,A
           ADD  A,28
           LD   (dienr),A         ; number to ascii

; left or right test
           LD   HL,(curpl)

w412       LD   A,%11101111
           IN   A,(254)
           CPL
           AND  3
           JR   Z,w412
           DEC  H
           JR   Z,persplay        ; human player

; AI-intelligence
; can I go to carrot clockwise        Y, go
; can I go to carrot anticlockwise    Y, go
; is anticlockwise field empty        N, go
; go clockwise
; no test on card available to take
           PUSH BC
compplay   LD   A,B
           CALL compai            ; set clockwise nr steps
           INC  L
           CP   (HL)
           POP  BC
           JR   Z,persplay+1      ; carrot clockwise
           PUSH BC
           LD   A,8
           SUB  B
           CALL compai            ; set anticlockwise nr steps
           INC  L
           CP   (HL)
           POP  BC
           JR   Z,setstep-2       ; carrot anticlockwise
           DEC  A
           LD   E,A
           ADD  A,A
           ADD  A,E
           LD   L,A
           BIT  0,(HL)            ; test 1st card on stack here
           JR   Z,setstep-2       ; cards here
           LD   A,1
persplay   DEC  A
           LD   D,1
           JR   Z,setstep         ; 0 pressed or AI-step 1/4
           LD   D,7               ; 1 pressed or AI-step 2/3
setstep    LD   A,D
           LD   (steps+1),A

steploop   PUSH BC

           CALL domove
           LD   (HL),A
           CALL mkbrd
```

```
            LD   HL,frames          ; movement delay
            LD   A,(HL)
            SUB  20
wfr         CP   (HL)
            JR   NZ,wfr

            POP  BC
            DJNZ steploop

; check position, card and how to handle
            LD   HL,(#4000+24)
            LD   B,L                ; B holds fieldnr

            LD   A,L                ; fetch current position
            CP   H                  ; compare against carrot
            LD   C,ca
            JR   Z,docarrot         ; handle carrotcard
            ADD  A,A
            ADD  A,L
            SUB  3
            LD   L,A
            LD   H,#40
            BIT  0,(HL)
            JR   NZ,wait            ; no cards here

            LD   C,(HL)             ; get current card

            CALL card2brd           ; show card on cursorpos
            JR   normalcard         ; not the carrot

docarrot    CALL card2brd
            LD   C,B
            LD   B,8
            CALL rnd
            LD   (#4000+25),A       ; new carrot position
            LD   B,A
            ADD  A,28
            LD   (carpos),A         ; show it in text
            LD   A,C
            CP   B
            LD   C,cy
            CALL NZ,card2brd        ; show only when diff new

; select opponent card after finding carrot
            LD   B,11
            CALL rnd
            LD   H,scrindex/256
            LD   B,2
            LD   L,A                ; start at random field
floop       LD   A,(HL)
            CP   em-1
            JR   NC,nfield          ; not in use in game

; valid cardtest
            CALL plindex            ; DE holds player
            LD   A,(HL)             ; fetch opponent field
fplpos      INC  E
            INC  E
            SUB  20
            JR   NC,fplpos          ; find matching player field
            LD   A,(DE)
            CP   (HL)               ; test if already is use
            JR   Z,nfield           ; when on own always true
```

```
            LD   A,(HL)            ; steal card
            LD   (HL),em           ; clear that field
            LD   (DE),A            ; set on player stack

            JR   wait

nfield      DEC  L                 ; goto next field
            JP   P,floop           ; not out of board, continue
            LD   L,11              ; set board around
            DJNZ floop             ; test nr of around
            JR   wait              ; here no card to steal

normalcard  CALL plindex           ; get player cards
            LD   A,C
fcpos       INC  E
            INC  E
            SUB  20
            JR   NC,fcpos          ; find position to fill
            LD   A,(DE)
            CP   C
            JR   Z,swapdeck        ; already in the deck

newcard     LD   A,C
            LD   (DE),A            ; set new card
            SET  0,(HL)            ; signal card taken

swapdeck    CALL Z,card2brd-2      ; alter visual card
            LD   B,3               ; swap max 3 times
test0       PUSH HL                ; save start
            LD   A,(HL)            ; get first card
            LD   D,H               ; point destination to same
            LD   E,L
            LD   C,H
            INC  HL                ; point to next
            LDI                    ; 2>1
            LDI                    ; 3>2
            LD   (DE),A            ; 1>3
            POP  HL                ; get start back
            DEC  B
            JR   Z,wait            ; 3 times moved = ready
            BIT  0,(HL)            ; test card is free
            JR   NZ,test0          ; if not, move again

wait        CALL plindex           ; get current player cards
            LD   B,6               ; test all 6 cards on sheet
wtest       INC  E
            INC  E
            LD   A,(DE)
            CP   em-1
            JR   NC,die6           ; if free, no winner
            DJNZ wtest             ; test 6 positions
            JP   restart           ; when here, player won

die6        LD   A,(dienr)
            CP   28+6              ; player threw "6"?

curpl       LD   A,0               ; who plays now?
            JR   Z,same            ; if 6, same player again
            XOR  3                 ; swap 1<>2 and 2<>1
same        JP   playloop          ; do next player

compai      LD   (steps+1),A
domove      LD   HL,#4000+24
```

```
steps       LD   B,0
            LD   A,(HL)             ; 1-8
fwbw        AND  7
            INC  A                  ; do 1 step on the board
            DJNZ fwbw
            RET

mkbrd       ADD  A,28
            LD   (posnr),A
            LD   B,8                ; 8 fields on board
            LD   DE,#4015           ; from stack 8 to stack 1
setboard    LD   C,em
            LD   HL,(#4000+24)      ; pawn and carrotposition
            LD   A,H                ; carrot stored here
            CP   B
            JR   Z,nocard-2         ; there is a card
            LD   A,(DE)
            BIT  0,A                ; still cards here?
            JR   NZ,nocard
            LD   C,cy               ; cardyes
nocard      LD   A,L                ; test on pawnposition
            CP   B
            JR   NZ,nopawn
            LD   C,pw
nopawn      CALL card2brd

            DEC  DE                 ; point to next stack
            DEC  DE
            DEC  DE
            DJNZ setboard           ; do 8 fields
;           RET    exit through plindex saves a byte

plindex     LD   A,(curpl+1)        ; get current player
            SUB  3
            LD   D,scrindex/256
            LD   E,A                ; now pos ok
            RET

n           EQU  27
dfile       DEFB 118                ; full lowres screen for init
            DEFB "P"-n,"L"-n,"A"-n,"Y"-n,"E"-n,"R"-n,0
plnr        DEFB 0
            DEFB 0,"T"-n,"H"-n,"R"-n,"O"-n,"W"-n,"S"-n,0
dienr       DEFB 0
            DEFB 118
            DEFB "C"-n,"A"-n,"R"-n,"R"-n,"O"-n,"T"-n,20
carpos      DEFB 29,0,"N"-n,"O"-n,"W"-n,20
posnr       DEFB 0
            DEFB 118

f1          EQU  board*256/256
; conversiontable from value to screenposition
boardtab    DEFB f1,f1+1
            DEFB f1+3,f1+5
            DEFB f1+7,f1+6
            DEFB f1+4,f1+2

board       EQU  $
;    f1 f2
; f8       f3
; f7       f4
;    f6 f5
vars        DEFB 128
last        EQU  $
```