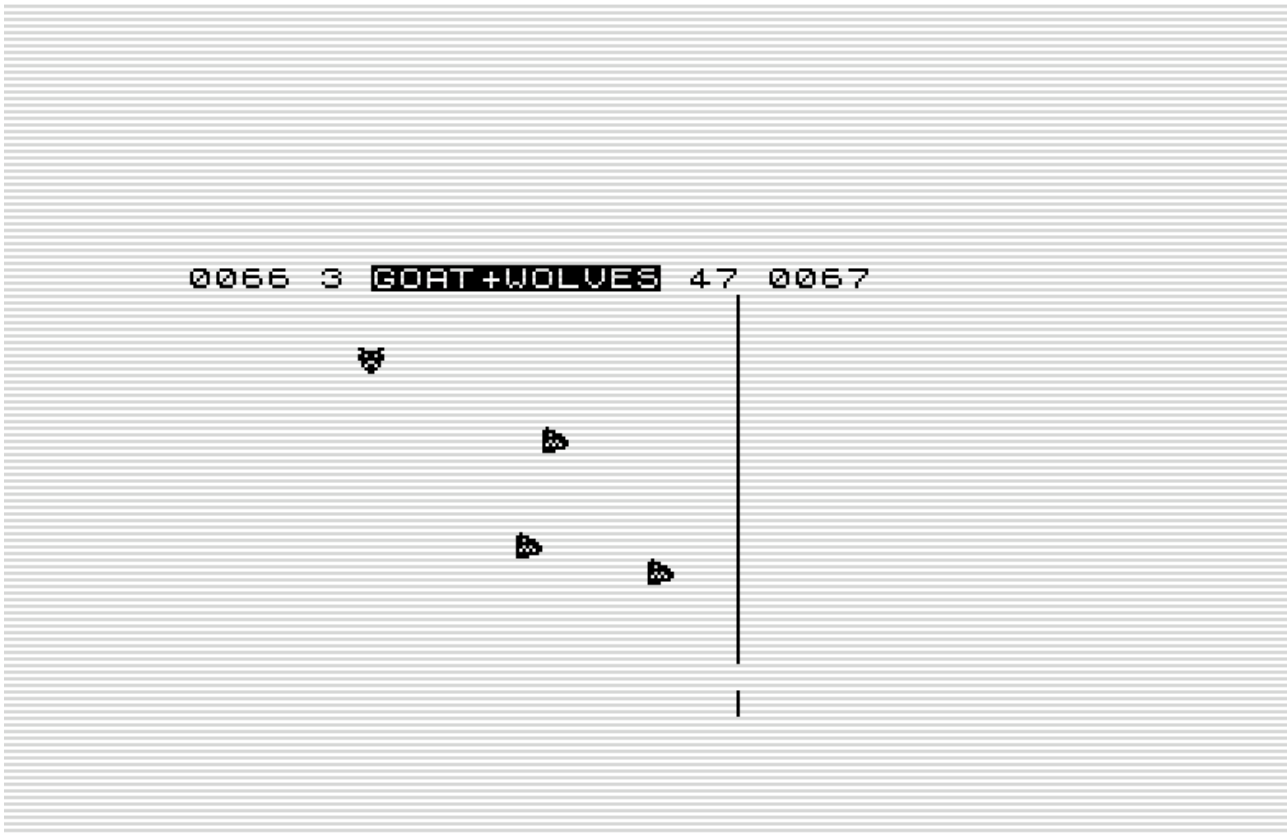


## Goat and wolves



**'67 is my year of birth. It is the chinese year of the goat. I thought of a game with a goat. Go through a field of wolves to get save was the idea for the game. In the end all kind of optimization was needed to get a good gameplay.**

```
; Goat and wolves
; Game 67 in 1K hires for the ZX81
```

```
rnd          EQU  nextlin
```

```
? * TORNADO *
```

```
ORG  #4009          ; #4009
DUMP 49161
```

```
basic        LD    D,#C0          ; preset for 48K bug
              JR    init0
```

```
DEFB 236,212,28    ; The BASIC
DEFB 126            ; fully placed over sysvar
DEFB 143,0,18       ; start to BASIC=#4009
```

```
eline        DEFW last            ; needed by loading
chadd        DEFW last-1
xptr         DEFW 0
stkbot       DEFW last
stkend       DEFW last
berg         DEFB 0
mem          DEFW 0
             DEFB 128
```

```
init1        JP    init          ; init can be anywhere
```

```

; all above reusable AFTER loading

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

; 11 bytes useable with LDIR, 10+5 when frames is skipped
nxtlin     DEFW basic           ; reusable after load

init0      LD    E,L             ; DE now #C0.L
           DEFB #26              ; HL now #40.L
flagx      DEFB 64

           XOR    A              ; intruptcounter reset
           EX     AF,AF'

taddr      DEFW 0                ; used by ZX81 on loading
           LD     B,4            ; copy >1K of code

frames     DEFW #DD01            ; used by ZX81, opcode IX
coprcc     LD     HL,hr          ; set IX to HR with opcode DD
sposn      JR     init1         ; continue to mainprog

cdflag     DEFB 64              ; used by ZX81

graphtab   DEFB #88,195,64,#98-1 ; show goat table
           DEFB #98,126,240,#A8-1
           DEFB #A8,219,220,#B8-1
           DEFB #B8,255,254,#C8-1
           DEFB #C8,102,215,#D8-1
           DEFB #D8,90,171,#B8-1
           DEFB #B8,60,254,#98-1
           DEFB #98,24,240

grtab2     DEFB #98,240,240,#98-1 ; no goat table
           DEFB #98,240,240,#A8-1
           DEFB #98,240,240,#B8-1
           DEFB #98,240,240,#C8-1
           DEFB #98,240,240,#D8-1
           DEFB #98,240,240,#B8-1
           DEFB #98,240,240,#98-1
           DEFB #98,240,240

field      PUSH BC
           INC    B
           LD     HL,lbuf1-17    ; B>0 C>=0
           LD     DE,17
fline      ADD    HL,DE          ; find right lbuf
           DJNZ   fline
           ADD    HL,BC          ; find position in lbuf
           POP    BC
           RET

; wolves screen must be between ..80 to ..ff
wolves     DEFB 64,64,64,64,64,64,64,64,64
           DEFB 64,64,64,64,64,64,64,64,16
w2         DEFB 240,240,240,240,240,240,240,240,240
           DEFB 240,240,240,240,240,240,240,240,16
w3         DEFB 220,220,220,220,220,220,220,220,220
           DEFB 220,220,220,220,220,220,220,220,16
w4         DEFB 254,254,254,254,254,254,254,254,254
           DEFB 254,254,254,254,254,254,254,254,16
w5         DEFB 215,215,215,215,215,215,215,215,215
           DEFB 215,215,215,215,215,215,215,215,16
w6         DEFB 171,171,171,171,171,171,171,171,171

```

```

        DEFB 171,171,171,171,171,171,171,16

; the display routine lowres and hires
hr      LD    HL,lowres+#8000    ; the lowres display
        LD    BC,#241          ; minimum needed #11
        LD    A,#1E
        LD    I,A
        LD    A,#FB
        CALL #2B5

hr00    LD    B,4                ; sync hires display
        DJNZ hr00

        LD    A,wolves/256
        LD    I,A                ; set highbyte

        LD    HL,retlbuf        ; return from highmem
        EXX
        LD    D,A                ; copy goat over wolves

        LD    H,graphtab/256    ; preset highbyte goatudg
        LD    (saveesp+1),SP
        LD    SP,#4000          ; displaystack is on sysvar
yposplay LD    B,#87+1          ; ypos add 255
        JR    bloop             ; retlbuf here save tstates

; screen placed here will set same higbyte text and nrbad
x       EQU    101
lowres  DEFB 118
score   DEFB 28,28,28,28,0
lives   DEFB 28,0
        DEFB "G"+x,"O"+x,"A"+x,"T"+x,149
        DEFB "W"+x,"O"+x,"L"+x,"V"+x,"E"+x,"S"+x,0
steps   DEFB 28,28,0
hiscore DEFB 28,28,34,35
        DEFB 118

retlbuf EXX                     ; here back from highmem
        DEC    E                 ; undo INC DE from LDI
        LDI    ; repair wolf
        DEC    C                 ; test end of 8 lines
        JR    NZ,cdelay         ; line filler and repair

bloop   LD    L,graphtab*256/256 ; default,we copy the goat
        DEC    B                 ; next line to show
        LD    C,31              ; 8 lines 16 copies 7 fillers
        LD    A,#87             ; first line to show
        CP     B                 ; test against show goat
        JR    NZ,setl           ; goat on other line
        JR    cloop             ; for same timing

setl     LD    L,grtab2*256/256  ; don't copy goat on this line

cloop   LD    E,(HL)             ; get x goat
        NOP                     ; filler
        INC    HL               ; point to data goat/no goat
        LDI    ; copy goat/no goat
        EXX
        LD    R,A               ; RET wil set R-pointer ok.
        RET                    ; use stack into highmen

cdelay  LD    A,(HL)             ; next line to show
        INC    HL               ; goto next x pointer
        DEC    SP               ; undo RET to highmem

```

```

        DEC SP
        DEC BC                ; filler, DE might get corrupt
        JR  cloop            ; BC only possible pair

; fixed end of HR-routine
savesp  LD  SP,0              ; repair stack
exit    CALL #292             ; back from intrupt
        CALL #220
        LD  IX,hr             ; set for next display
        JP  #2A4

eog     LD  DE,hiscore-1      ; pointer to hiscore
        LD  HL,score-1        ; pointer to current score
        LD  BC,5              ; lenght to test
fihi    INC  HL                ; next digit in score
        INC  DE                ; next digit in hiscore
        DEC  C                ; digit less to copy
        LD  A,(DE)            ; get hiscore digit
        CP  (HL)              ; test against score
        JR  Z,fihi            ; still the same
        CALL C,#19F9          ; new hiscore through ROM

start   LD  A,(lastk)         ; game over, wait for
        SUB %10111111         ; newline
        JR  NZ,start

        LD  L,score*256/256
clsc    LD  (HL),28            ; reset digit
        INC  HL
        CP  (HL)              ; space after score?
        JR  NZ,clsc           ; clear full score

        INC  A
        LD  (nrbad+1),A        ; set first wolf

        INC  HL                ; next is lives
        LD  (HL),32            ; set 4 lives = 3 to play

dead    LD  L,lives*256/256    ; live lost
        DEC  (HL)
        LD  A,(HL)
        CP  28
        JR  Z,eog             ; nothing left, game over

newround CALL clearscr         ; erase screen

        LD  (deadbywolf+1),A    ; signal not hit

        LD  C,B
        CALL showgoat          ; also no wolf here now

; store xy badies in memory
nrbad   LD  B,0                ; display nr wolfs
        LD  DE,badxytab        ; table xy wolves
makewolf PUSH BC                ; save counter
        PUSH DE                ; save pointer
nxtrnd  CALL rnd               ; get rnd 0-15
        CALL rnd               ; set rnd y, get rnd 0-15
        LD  C,A                ; set rnd X
        CALL field
        BIT 6,(HL)             ; test wolf shown
        JR  Z,nxtrnd           ; not on used field
        LD  (HL),B             ; set next wolf to show
        POP  DE                ; get pointer

```

```

CALL setxy                ; store xy in table

POP BC
DJNZ makewolf             ; set all wolves

wnl
LD HL,lastk
LD A,(HL)
CP 191
JR NZ,wnl                 ; wait for newline

DEC HL
LD (HL),H                 ; set timer

playloop
LD HL,lastk-1             ; stepcounter over sysvar

LD A,H
LD (15*17-2+lbuf1),A ; set gate open

DEC (HL)

LD A,(HL)                 ; get remaining steps
LD HL,steps               ; preset H for lives too
JR Z,dead                 ; test end of timer

deadbywolf
JR dead                   ; altered for test

setten
LD (HL),27
INC (HL)
SUB 10
JR NC,setten              ; make 10 digit visible
INC HL
ADD A,38                  ; set unit
LD (HL),A
CALL showgoat

LD L,frames*256/256      ; delay to play
LD A,(HL)
SUB 12
CP (HL)
JR NZ,wfr

LD (oldxy+1),BC          ; save old X Y player

PUSH BC

CALL clearscr             ; clear screen

LD BC,(lastk)             ; get lastkey
LD A,C                   ; port to A, no key = 255
INC A                     ;
CALL NZ,#7BD              ; get keycode
POP BC
PUSH BC                   ; save old xy
CP 10                     ; Q pressed?
JR NZ,t2
DEC B
CP 5                       ; A pressed?
JR NZ,t3
INC B
SUB 26                    ; O pressed?
JR NZ,t4
DEC C
INC A                     ; P pressed?
JR NZ,testmove

```

```

testmove    INC    C
            LD     HL,nrbad+1      ; preset HL
            LD     A,15            ; test out of screen
            CP     B
            JR     C,false         ; undo move
            CP     C
            JR     C,false         ; undo move
            POP    DE              ; drop old bc
            JR     NZ,valid        ; on exit column
            DEC    C               ; undo step right
            LD     A,B
            CP     C               ; test row
            JR     NZ,valid        ; gate reached

            CP     (HL)            ; test max wolves reached
            JR     Z,skipinc
skipinc     INC     (HL)           ; add a wolf
            LD     A,(lastk-1)     ; get remaining time
            LD     B,A             ; set time as counter
addsc       LD     L,score*256/256+4
ten         DEFB 17
            LD     (HL),28
            DEC    HL
            INC    (HL)
            LD     A,(HL)
            CP     38
            JR     Z,ten
            DJNZ  addsc            ; add all points
            JP     newround

false       POP    BC             ; get unchanged XY player
; move baddies
valid       LD     (xynow+1),BC    ; save XY player
            LD     B,(HL)          ; get nr wolfs
            LD     DE,badxytab    ; get XY pointer
movewolf    PUSH   BC
            LD     A,(DE)          ; get X and Y wolf
            RRCA
            RRCA
            RRCA
            RRCA
            AND    15
            LD     B,A             ; set Y
            LD     A,(DE)
            AND    15
            LD     C,A            ; set X

            CALL  wolfongoat      ; test hit before move

; oldxy for movement
oldxy       LD     HL,0
            LD     A,H             ; get old y player
            CP     B               ; test y current wolf
            LD     A,L             ; preload old x player
            JR     Z,movex         ; horizontal view, move x
            CP     C               ; test x current wolf
            JR     NZ,setwolf      ; no view on goat
; view vertical
movey       LD     A,H
            CP     B               ; position wolf vs player
            SBC    A,A             ; 0 vs -1
            ADD    A,A             ; 0 vs -2
            INC    A               ; 1 vs -1
            ADD    A,B             ; calculate new Y wolf

```

```

        LD    B,A                ; save Y
        JR    setwolf

movex   CP    C                ; same with x wolf
        SBC  A,A
        ADD  A,A
        INC  A
        ADD  A,C
        LD   C,A

setwolf CALL setxy

        CALL wolfongoat        ; test hit after move
        PUSH DE
        CALL field
        LD   (HL),B            ; show wolf
        POP  DE
        POP  BC
        DJNZ movewolf          ; move and show all wolves

xynow   LD    BC,0              ; get XY player back
        JP    playloop

wolfongoat LD HL,(xynow+1)      ; current xy player
        SBC  HL,BC              ; take off xy wolf
        RET  NZ                ; not hit is return

jrdead  LD    A,0
        LD   (deadbywolf+1),A   ; set a number in deadbywolf
;        RET                    ; show goat over wolf

showgoat CALL field

        LD   (HL),B            ; show field player

        LD   HL,graphtab
        LD   A,(HL)             ; get old pointer+x
        SUB  #88                ; take off first pointer
        LD   E,A                ; old x to e
        LD   D,8                ; 8 lines to write pointers
setgoat LD   A,(HL)             ; get old pointer+x
        SUB  E                  ; take off old x
        ADD  A,C                ; add new x
        LD   (HL),A            ; write new pointer
        CALL #7B8              ; point to next line
        DEC  D                  ; next line to do
        JR   NZ,setgoat

        LD   A,B                ; get Y goat
        ADD  A,#88              ; add pointer
        LD   (yposplay+1),A     ; set goatdisplay on
;        RET                    exit through setxy DE=ROM

setxy   LD    A,B
        ADD  A,A
        ADD  A,A
        ADD  A,A
        ADD  A,A
        ADD  A,C                ; A = yyyxxxxx
        LD   (DE),A            ; X and Y stored in 1 byte
        INC  DE
        RET

clearscr LD HL,lbuf1

```

```

        LD    A,16                ; use A to clear for later
cls      LD    B,15
ccline  LD    (HL),#64            ; display off
        INC    HL
        DJNZ   ccline
        LD    (HL),B              ; but show fence
        INC    HL
        INC    HL
        DEC    A
        JR     NZ,cls
        RET

badxytab DEFS 14                  ; 14 x and y from wolves

size     EQU   16*17

st        EQU   26                ; stack size

space     EQU   #4400-size-st-$

        DEFS  space

; executable code on the stack, 1 time only
stackcode LDIR                    ; copy all linedisplay buffers
        JP     eog                ; start through end of game

        DEFS  st-5                ; SP-filler: size SP 26 bytes

lbuf1     DEFW #4040,#4040         ; on load all fields invisible
        DEFW #4040,#4040         ; program will determine
        DEFW #4040,#4040         ; which fields need to show
        DEFW #4040,#4040         ; the UDG on that field
        JP     (HL)              ; return lowmemory

screen    EQU   $
; in fact 2nd line of screen will start here, 1st is lbuf1

; initialization code on screen is done before
; first screen is called to be drawn

init      LDIR                    ; repair 48K bug
        LD    HL,lbufstack        ; get displaystack from screen
        LD    DE,#4000            ; destination: sysvar
        LD    C,34                ; copy stack to now free mem
        LDIR                    ; save 34 bytes

        LD    A,(deadbywolf+1)    ; save the JR on dead
        LD    (jrdead+1),A        ; to be set back when needed

        LD    HL,rnd2             ; the randomroutine
        LD    DE,nxtlin           ; can go over sysvar too
        LD    C,18                ; saving another 18 bytes
        LDIR

        LD    HL,lbuf1            ; get displayline
        LD    SP,HL               ; move SP from end of RAM
        LD    DE,screen           ; set is behind first line
        LD    BC,size-17          ; copy it 15 times
        JP    stackcode           ; copy must be done elsewhere

rnd2      LD    B,A                ; random Y is set over sysvar
        LD    HL,lastk-2          ; seed pointer
        LD    A,(HL)              ; get seed
        RRCA                      ; a=a/2

```



```

RRCA                ; a=a/2
XOR 63              ; swap low bits
ADD A,(HL)          ; add seed
DEFB 17             ; hide frames in DE
DEFW 65535          ; frames used by zx81
ADD A,E             ; add framecounter
LD (HL),A           ; save new seed
AND 15              ; we only need 0-15
RET                ; exit rnd

lbufstack DEFW lbuf1+#8000
          DEFW lbuf1+#8000+17
          DEFW lbuf1+#8000+34
          DEFW lbuf1+#8000+51

          DEFW lbuf1+#8000+68
          DEFW lbuf1+#8000+85
          DEFW lbuf1+#8000+102
          DEFW lbuf1+#8000+119

          DEFW lbuf1+#8000+136
          DEFW lbuf1+#8000+153
          DEFW lbuf1+#8000+170
          DEFW lbuf1+#8000+187

          DEFW lbuf1+#8000+204
          DEFW lbuf1+#8000+221
          DEFW lbuf1+#8000+238
          DEFW lbuf1+#8000+255

          DEFW savesp          ; exit screen by RET

vars      DEFB 128
?
last      EQU $

```