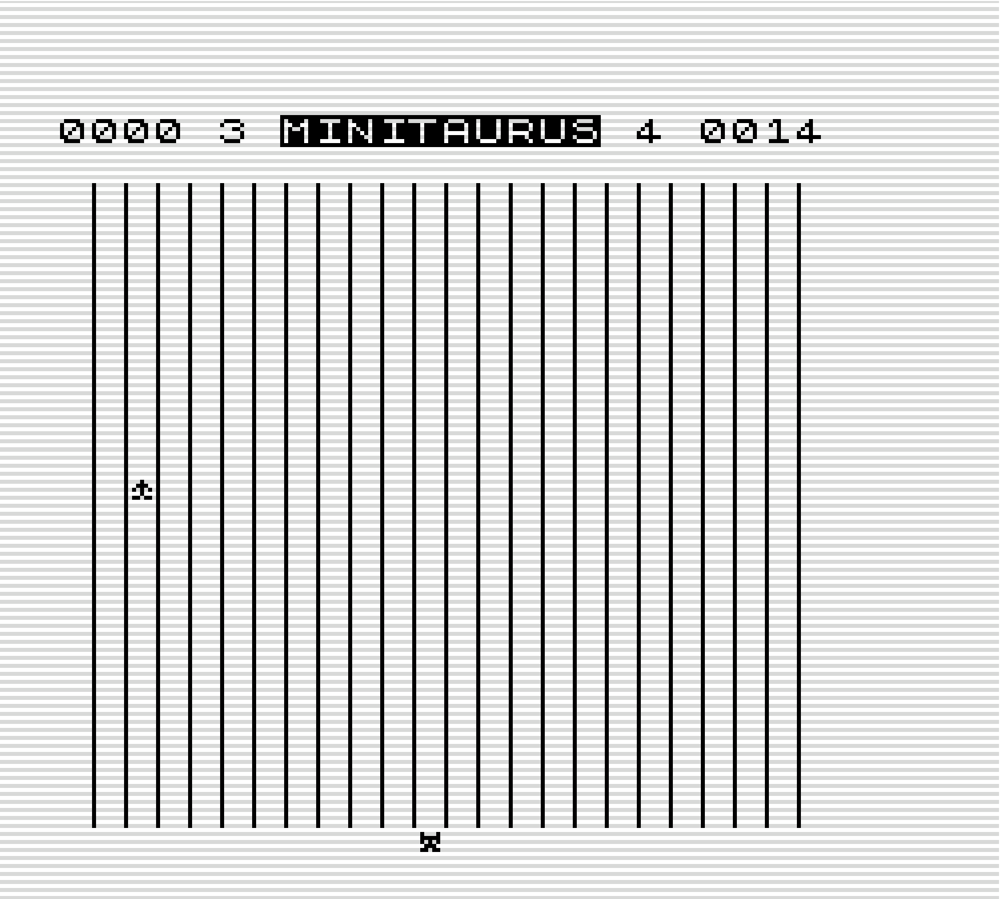
Minitaurus



```
0000 3 MINITAURUS 4 0014
```

I always wanted to make a computergame of this. My attempt on the ZX Spectrum was not the way I wanted and it swapped too slow. With the knowledge of the ZX81 display I realized it could be done on a ZX81. And it is exactly as I wanted it to be.

```
; MINI-taurus maze (free after King Minos minotaurus maze)
; This game is based on a game we played during PE
; in elementary school. 25 children made the 5x5 maze
; and 1 kid was chasing an other while the teacher
; whistled the mazechange

? * TORNADO *

           ORG  #4009             ;#4009
           DUMP 49161

basic      LD   B,5               ; 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
```

```
; all above reusable AFTER loading

lastk      DEFB 255,255,255       ; used by ZX81
margin     DEFB 55                ; used by ZX81
nxtlin     DEFW basic             ; reusable after load

init0      XOR  A                 ; delay intrupts by
           DEFB 254               ; CP n ; skip flagx
flagx      DEFB 0

           EX   AF,AF'            ; intruptcounter reset
           DEFB 17                ; LD DE,nn ; skip taddr

taddr      DEFW 3213              ; used by ZX81
           JR   init1             ; continue to REAL init

frames     DEFW 65535             ; used by ZX81
coord      DEFB 0,0               ; useable
prcc       DEFB 188               ; used by ZX81
sposn      DEFB 33,24             ; used by ZX81
cdflag     DEFB 64                ; used by ZX81

em         EQU  empty*256/256
lbuf       LD   R,A               ; hires display
empty      DEFW 0,0,0,0,0,0       ; top and bottom empty
           DEFW 0,0,0,0,0,0       ; displayline also linebuffer
           LD   B,B               ; filler
           JP   NZ,low            ; end of screen test, 48K bug
           JP   screxit

vl         EQU  vert*256/256      ; the vertical lines
vert       DEFB 1,1,1,1,1,1
           DEFB 1,1,1,1,1,1
           DEFB 1,1,1,1,1,1
           DEFB 1,1,1,1,1,0

hr         LD   HL,lowres+#8000   ; the lowres display
           LD   BC,#209           ; minimum lines needed
           LD   A,#1E
           LD   I,A
           LD   A,#FB
           CALL #2B5              ; display lowres

           LD   B,8               ; sync hires display
hr00       DJNZ hr00

           LD   B,176             ; 176 lines
           LD   D,#40             ; pointer to table
           LD   HL,screen         ; pointer to UDG-screen

low        LD   A,B               ; test current line
           CP   (HL)              ; is it UDG-line
           JR   Z,doscreen        ; if so display UDG
           SUB  8                 ;
           CP   161               ; bottom and top in 1 test
           JR   NC,topbot         ; display top or bottom
           AND  7                 ; select right line
           LD   E,A               ; save tableindex
           LD   A,D               ; set highbyte
           LD   I,A
           LD   A,(DE)            ; get lowbyte per line
           NOP                    ; filler
           DEC  B                 ; 1 line less
           JP   lbuf+#8000        ; do display of line

ho         EQU  horl*256/256      ; horizontal line data
horl       DEFB 000,255,255,255,255,255
           DEFB 255,255,255,255,255,255
           DEFB 255,255,255,255,255,255
           DEFB 255,255,255,255,255,000

doscreen   LD   A,H               ; highbute in 2 ranges
           LD   I,A               ; set is needed on each line
           LD   A,L
           LD   E,A               ; save lowbyte
           ADD  A,25
           LD   L,A               ; point to next line
           CP   (HL)              ; filler
           NOP                    ; filler
           LD   A,E               ; get current line
           INC  A                 ; skip line index
           DEC  B                 ; also 1 line less
           JP   NZ,lbuf+#8000     ; do display, but not final

; fixed end of HR-routine
screxit    CALL #292              ; back from intrupt
           CALL #220
           LD   IX,hr
           JP   #2A4

topbot     LD   A,#40             ; set highbyte
           LD   I,A
           LD   A,em              ; get emptypointer
           CP   (HL)              ; filler
           DEC  B                 ; 1 line less
           JP   lbuf+#8000        ; do display

endofgame  LD   HL,score-1        ; hiscore check
           LD   DE,hiscore-1
           LD   BC,5
fhigh      DEC  C
           INC  HL
           INC  DE
           JR   Z,start
           LD   A,(DE)
           CP   (HL)
           JR   Z,fhigh
           JR   NC,start
           LDIR

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

playgame   LD   HL,#1C1C          ; reset score
           LD   (score),HL
           LD   (score+2),HL

           LD   A,32              ; set "4" lives
           LD   (lives),A

dead       LD   HL,lives          ; take of 1 so start with 3
           DEC  (HL)
           LD   A,(HL)
           CP   28
           JR   Z,endofgame       ; no lives left
```

```
nextman    LD   B,22
           CALL rnd
           LD   (xycomp+1),A      ; random X of computer
           CALL rnd
           ADD  A,A
           ADD  A,A
           ADD  A,A              ; Y goes per 8 lines
           INC  A
           LD   (xycomp+2),A      ; random Y of computer

           LD   A,1              ; timer set to end
           LD   (timecnt+1),A

           LD   A,6              ; scoretimer 1 more
           LD   (sectime+1),A

xyplay     LD   HL,#0910
xycomp     LD   DE,#A901

swapdehl   EX   DE,HL
           LD   A,D
           CP   H
           JR   C,swapdehl
           PUSH HL               ; save smallest
           SBC  HL,DE
           JR   Z,point1

           LD   HL,#4303          ; start of screen
           CALL clrudg            ; do clrudg with highest A
           POP  AF               ; smallest to A
           LD   L,128            ; second screenpart
           CALL clrudg            ; do clrudg with smallest Y

           LD   HL,(xyplay+1)     ; regain pointers
           LD   DE,(xycomp+1)
           LD   BC,#1015          ; UDG1 and UDG2
           LD   A,D
           CP   H
           LD   A,C              ; second pointer
           JR   NC,dode
           EX   DE,HL
           LD   A,B              ; save other pointer
           LD   B,C              ; set correct B
dode       PUSH HL               ; save smallest
           PUSH AF               ; save flags and next pointer
           LD   HL,screen         ; first always on start
           CALL setitem
           POP  AF
           LD   B,A              ; next item to B
           JR   NZ,setitem2       ; HL is set ok
           LD   E,3              ; back tostart
setitem2   EX   DE,HL
           POP  DE
           CALL setitem

           LD   HL,frames         ; speed delay
           LD   A,(HL)
           SUB  5
wfr        CP   (HL)
           JR   NZ,wfr

           LD   BC,(lastk)        ; get keypressed
           LD   A,C
           INC  A
           CALL NZ,#7BD           ; translate if keypressed
           LD   DE,xyplay+2       ; point to x and y player
           LD   BC,(xyplay+1)     ; get x and y player
           CALL checkdir          ; check valid move

           XOR  A                 ; reset carry
           LD   HL,(xyplay+1)     ; test man found after
           LD   DE,(xycomp+1)     ; movement of player
           SBC  HL,DE
point1     JR   Z,sectime

compdir    LD   B,4               ; a random directiom for
           CALL rnd               ; computer to go to
           DEC  A
oppdir     CP   0                 ; move back not allowed
           JR   Z,compdir
           PUSH AF                ; save direction
           ADD  A,init0*256/256
           LD   L,A
           LD   H,#40
           LD   A,(HL)            ; from direction to "keypress"
           LD   DE,xycomp+2       ; point to x and y computer
           LD   BC,(xycomp+1)     ; get x and y computer
           CALL checkdir          ; test move
           POP  HL                ; get direction from stack
           CP   D                 ; test on valid move
           JR   NZ,compdir        ; redo move until valid
           LD   A,H               ; get valid move
           XOR  1                 ; calculate opposite direction
           LD   (oppdir+1),A      ; save as not allowed

timecnt    LD   A,0               ; some time for each score
           DEC  A
settime    LD   (timecnt+1),A
           JR   NZ,timer

           LD   HL,sectime+1      ; time passed, lower score
           LD   A,(HL)
           ADD  A,27
           LD   (time),A          ; display less score
           DEC  (HL)              ; fysical decrease in score
           JP   Z,dead            ; out of score=loss of live
           LD   A,40              ; set new timer
           JR   settime

sectime    LD   B,0               ; remaining score
addscore   LD   HL,score+4        ; added to score
           DEFB #3A
tens       LD   (HL),28
           DEC  HL
           INC  (HL)
           LD   A,(HL)
           CP   38
           JR   Z,tens
           DJNZ addscore
           JP   nextman

timer      LD   A,16              ; the mazetimer
           DEC  A
           LD   (timer+1),A
           JP   NZ,xyplay         ; no swap of direction

swapdir    LD   HL,#4000          ; swap the direction
           LD   DE,#4008          ; of the maze
           LD   B,E
           LD   C,H
swap       LD   A,(DE)
           LDI
           DEC  HL
           LD   (HL),A
           INC  HL
           DJNZ swap

           INC  B                 ; calculate new mazetimer
           CALL rnd
           ADD  A,A
           ADD  A,A
           ADD  A,A
           ADD  A,A
           JR   timer+2           ; set timer

rnd        LD   DE,0              ; seed
           LD   HL,(frames)       ; timer from computer
           ADD  HL,DE
           DEC  HL                ; point to new start
           LD   A,H
           AND  #1F
           LD   H,A               ; but stay in ROM
           LD   (rnd+1),HL        ; save new seed
           LD   A,(HL)            ; get random value
frnd       SUB  B                 ; calculate within range
           JR   NC,frnd
           ADC  A,B
           RET

checkdir   LD   HL,init0          ; keys set here
           CP   (HL)
           CALL Z,up
           INC  HL
           CP   (HL)
           INC  HL
           CALL Z,down
           CP   (HL)
           INC  HL
           CALL Z,right
           CP   (HL)
           RET  NZ

left       LD   A,C
           SUB  2
           LD   C,A
right      LD   A,C
           DEC  DE
           CP   24
           LD   A,#FF             ; false value
           RET  NC                ; out of screen
           INC  C
           LD   A,B
           SUB  8
           CP   161
           JR   NC,okmove1        ; border of maze allowed
           LD   A,(#4001)         ; within maze
           SUB  vl                ; test right wall
           RET  Z                 ; vert. wall on hor. move
okmove1    LD   A,C
           LD   (DE),A            ; set new X
exit       LD   A,D               ; signal move succesfull
           RET
```

```
up         LD   A,B
           ADD  A,16
           LD   B,A
down       LD   A,B
           SUB  8
           LD   B,A
           CP   176
           RET  NC                ; out of screen
           LD   A,C
           SUB  2
           CP   22
           JR   NC,okmove         ; border of maze allowed
           LD   A,(#4001)         ; within maze
           SUB  em                ; test lack of vert. wall
           RET  Z                 ; no vert. wall=illegal move
okmove     LD   A,B
           LD   (DE),A            ; set new Y
           LD   A,D               ; signal move succesfull
           RET                    ; also impossible keyvalue

; set background and item
; DE is where, B is graphic
setitem    LD   A,D
           LD   (yposudg+1),A     ; save Y-pos for setting
           SUB  8
           CP   161
           SBC  A,A
           PUSH HL
           LD   HL,#4000
           LD   L,(HL)
           AND  (HL)              ; no background on border
           LD   C,E               ; xpos to C
           POP  DE
           LD   L,B               ; pointer to graphic
           LD   B,5
yposudg    LD   A,0
           ADD  A,B
           LD   (DE),A            ; save linenumber
           LD   A,E
           ADD  A,C
           LD   E,A
           LD   A,(DE)            ; get current background
           OR   (HL)              ; add udg
           LD   (DE),A            ; set all on screen
           LD   A,E
           SUB  C
           ADD  A,24              ; point to end of line
           LD   E,A
           EX   DE,HL
           RES  0,(HL)            ; reset border "background"
           EX   DE,HL
           XOR  A
           INC  DE
           LD   (DE),A            ; set impossible nextline
           INC  HL
           DJNZ yposudg
           RET

clrudg     SUB  8
           CP   161
           SBC  A,A               ; top or bottom test
           PUSH HL
           LD   HL,#4001
           LD   L,(HL)
           AND  (HL)
           POP  HL
clrloop    LD   (HL),A            ; set correct background
           INC  L
           JR   NZ,clrloop
           RET

lowres     DEFB 118
s          EQU  128-27

score      DEFB "U"+s,"D"+s,"R"+s,"L"+s,0
lives      DEFB "U"+s,0
           DEFB "M"+s,"I"+s,"N"+s,"I"+s
           DEFB "T"+s,"A"+s,"U"+s,"R"+s,"U"+s,"S"+s,0
time       DEFB 28,0
hiscore    DEFB 28,28,28,28
           DEFB 118

; part if init becomes stack
; this will happen only AFTER code is run
; this is done by delaying intrupts
; The intrupt will use the stack in needed size
init       LD   IX,hr             ; Hires mode
           LD   SP,screen
           LD   H,#3F             ; #3fxx
           LD   D,#BF             ; #bfxx
           LD   E,L
           LDIR                   ; repair 48K bug
           LD   DE,#4000
           LD   HL,tab
           LD   C,26              ; table + udg
           LDIR
           LD   HL,start
           PUSH HL                ; start on stack
           LD   HL,score
           JP   keydef

space      EQU  #4303-$
           DEFS space

screen     EQU  $
; a nice loadingscreen in hires, why not????
           DEFB 98
           DEFB 32,200,74,13,44
           DEFB 185,43,41,129,136,156
           DEFB 228,164,202,204,12,238
           DEFB 202,164,232,209,49,59
           DEFB 17
           DEFB 97
           DEFB 112,168,170,10,138
           DEFB 18,170,170,24,85,132
           DEFB 42,170,170,170,10,136
           DEFB 170,170,136,170,169,213
           DEFB 31
           DEFB 96
           DEFB 168,200,228,10,170
           DEFB 147,171,41,16,148,136
           DEFB 42,238,160,172,12,204
           DEFB 192,202,200,171,169,153
           DEFB 21
           DEFB 95
           DEFB 32,136,164,10,170
           DEFB 146,170,168,153,20,136
           DEFB 170,170,160,170,10,136
           DEFB 128,170,136,170,169,221
           DEFB 14
           DEFB 94
           DEFB 216,142,164,10,170
           DEFB 146,146,147,1,201,200
           DEFB 68,170,160,202,12,238
           DEFB 128,164,238,170,169,51
           DEFB 27

; impossible linenumber to hide
; the redefinecode of the screen
           DEFB 255

keydef     LD   E,init0*256/256
redef      LD   A,(lastk)
           INC  A
           JR   NZ,redef          ; wait for a key up
waitkey    LD   BC,(lastk)        ; get current keycode
           LD   A,C
           INC  A
           JR   Z,waitkey         ; wait for a key down

           PUSH HL
           PUSH DE
           CALL #7BD              ; translate keycode to ASCII
           POP  DE
           POP  HL
           LD   (DE),A            ; save ASCII-value
           INC  DE
           INC  HL
           LD   A,(HL)            ; set new direction to ask
           LD   (lives),A
           OR   A
           JR   NZ,redef          ; do all directions
           RET                    ; goto start of game

; displaytable for HOR/VER lines
; swap on display
; table and UDG are copied over sysvar
tab        DEFB ho,em,em,em,em,em,em,em
           DEFB vl,vl,vl,vl,vl,vl,vl,vl

udg        DEFB 16,56,84,16,108   ; man
           DEFB 68,124,84,56,108  ; Mini-taurus

vars       DEFB 128
?
last       EQU  $
```