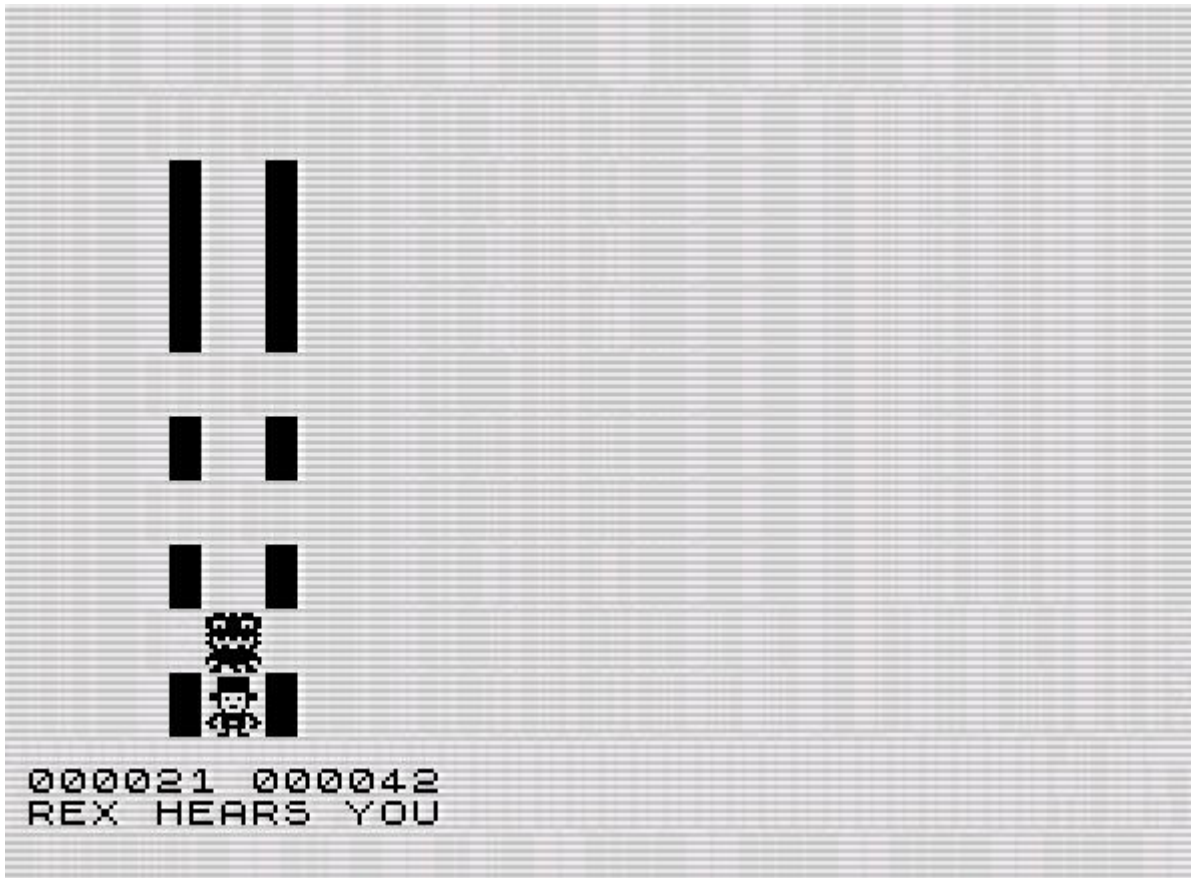


2D Monster Maze



A tribute to JK Greye's 3D Monster Maze. The iconic gameplay can be done in just 1K. Well, at the cost of the 3D view. With the topview in a maze you can play the same game. Also the mist of time is reduced to a few seconds to generate the maze. Rex is hungry and he will not rest until he is fed. Find the food or become the food. The graphics are displayed on the screen when needed. The displaymethod resembles the display of the car from Police Patrol. Although more graphics can be displayed in this game.

```
; 2D Monster Maze
; Rex' lair in 1K
; The same gameplay with the same thrill.
; If the suspense won't kill you, Rex will.
```

```
? * TORNADO *
```

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

```
JP     init
```

```
d_file    DEFW dfile
dfcc      DEFW dfile+1
var       DEFW vars
dest      DEFW 0
eline     DEFW last
chadd     DEFW last-1
xptr      DEFW 0
stkbob    DEFW last
```

```

stkend      DEFW last                ; memory above reused for data

berg        DEFB 0
            DEFW 0
            DEFB 0
            DEFB 2
            DEFW 1

lastk       DEFB 255,255,255        ; used by ZX81

margin      DEFB 55
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

movetab     DEFB up*256/256
            DEFB right*256/256
            DEFB down*256/256
            DEFB left*256/256

down        INC B
            CALL field
            RET NC
            DEC B
            RET

up          DEC B
            CALL field
            RET NC
            INC B
            RET

right       INC C
            CALL field
            RET NC
            DEC C
            RET

left        DEC C
            CALL field
            RET NC
            INC C
            RET

startgame   LD HL,score-1           ; test for hiscore
            LD DE,hiscore-1
            LD BC,7

seekhi      INC HL
            INC DE
            DEC C
            JR Z,w4enter
            LD A,(DE)

```

```

        CP      (HL)
        JR      Z,seekhi
        JR      NC,w4enter
        LDIR                                ; set hiscore

w4enter  LD      A,(lastk)
        CP      %10111111
        JR      NZ,w4enter                ; start with NEWLINE

        LD      HL,score
        LD      B,6
ressc    LD      (HL),28                    ; reset score
        INC     HL
        DJNZ    ressc

restart  LD      HL,score+3                ; each food is 100 points
        CALL    NZ,setscore                ; not called on start of game
        LD      B,5                        ; 2 sec dealy for msg
startdelay CALL    delay
        DJNZ    startdelay

start    LD      B,34
        LD      HL,maze
clmaze   LD      (HL),255
        INC     HL
        DJNZ    clmaze                    ; maze is cleared

eoy      CALL    rnd                        ; start at random Y for x-loop
        LD      B,A
        LD      DE,1
        LD      C,E

setline  PUSH    DE
        CALL    field
        LD      A,(HL)
        AND     D                        ; erase field
        LD      (HL),A
        POP     DE
        LD      A,B
        ADD     A,D
        AND     15
        LD      B,A
        JR      Z,eoy                    ; end of vertical move found
        LD      A,C
        ADD     A,E
        AND     15
        LD      C,A
        JR      NZ,setline                ; do next field in line
eox      CALL    rnd                        ; end of hor move reached
        LD      C,A                        ; new random X
        LD      DE,#100
        LD      B,D
linecnt  LD      A,0
        INC     A
        AND     7
        LD      (linecnt+1),A            ; do 8 hor and 8 vert lines
        JR      NZ,setline
        LD      (movedir+1),A            ; start directionview
; maze ready, 0 in field is access

        CALL    wallhit                    ; find start of rex
        LD      (rexxy+1),BC

```

```

CALL wallhit          ; find foodposition
LD (foodxy+1),BC

startxy CALL wallhit          ; find start for player
LD HL,(rexxy+1)
SBC HL,BC             ; do not start on rex
JR Z,startxy         ; if so, find new start

;make screen
loop PUSH BC           ; save for after display
LD HL,datadisp+50     ; begin at end, hatman place
PUSH HL
LD HL,hatman*256+62   ; "LD A,hatmanudg"

scrloop LD A,10         ; max view 10 deep
LD (setloop+1),A
LD (defudg),HL       ; set LD A
PUSH BC             ; store xy-coordinates
LD A,1             ; move right
CALL movedir
POP BC
POP DE
DEC DE             ; to right wall
SBC A,A           ; 0 or 255
LD (DE),A         ; set wall or passage
DEC DE           ; udg data1
DEC DE           ; udg data2
DEC DE           ; left wall index
PUSH DE
PUSH BC
LD A,255         ; move left
CALL movedir
SBC A,A
POP BC
POP DE
LD (DE),A       ; full loop done
DEC DE

defudg LD A,0     ; hatman or space
LD (DE),A

rexxy XOR rex*256/256 ; for test always 0, reset C
LD HL,#C0A
SBC HL,BC
JR Z,setudg     ; rex must be set
AND A          ; reset carry

foodxy LD A,food*256/256
LD HL,#C02
SBC HL,BC
JR NZ,noudg

setudg LD (DE),A   ; set correct udg
noudg PUSH DE
XOR A      ; move forward
CALL movedir

setloop LD A,0
JR C,ready ; wall hit
LD HL,62   ; default to space
DEC A
JR NZ,scrloop ; do all views
POP DE
JR filler

ready POP HL      ; display final wall
DEC A          ; make rest final wall and
JR Z,fill     ; spaces

```

```

        DEC HL
        SBC A,A                ; 255
        LD (HL),A             ; wall
        DEC HL
        DEC HL
        DEC HL
        LD (HL),A
fill    EX DE,HL
        LD A,wall*256/256
fillloop DEC DE
        LD (DE),A
filler  XOR A
        LD HL,datadisp-1
        SBC HL,DE
        JR C,fillloop

; determine message, compute number
        LD B,3                ; nr of message
        POP DE
        LD A,(datadisp+45)
        CP hatman*256/256
        JR NC,nxtmsg
        LD A,rex*256/256
        LD (datadisp+45),A    ; repair graphic
        LD HL,startgame
foodmsg PUSH HL                ; store RETURN
msgfnd  PUSH DE                ; store BC
        LD HL,texting
fcr     LD A,(HL)
        INC HL
        CP #76
        JR NZ,fcr
        DJNZ fcr
        LD C,14
        LD DE,message
        LDIR                  ; copy message to screen
        INC C
        POP BC                ; fetch XY
        RET

nxtmsg  JR Z,nofood
        INC B
        LD HL,restart
        JR foodmsg
        JR nxtmsg
nofood  DEC B
        LD HL,(rexxy+1)
        LD A,D
        CP H
        LD A,E
        LD C,0
        JR Z,nearby
        INC C
        CP L
        LD A,D
        LD L,H
        JR NZ,nextmsg

; nearby will also make REX move to you.
nearby  CP L                    ; from Y E<>L from X D<>H
        SBC A,A                ; left or right
        ADD A,A
        INC A                  ; new direction for rex
        ADD A,C                ; add lr/ud index

```

```

nextmsg      LD    (dirrex+1),A
msgok        XOR    A
             LD    (movenr+1),A    ; do atleast 4 moves to player
             INC    B                ; undo dec
             DEC    B
             CALL   msgfnd
             CALL   delay

             LD    A,(lastk)
             LD    E,255
             CP    %11110111        ; 1-5 left turn
             CALL   Z,altervwpl
             LD    E,1
             CP    %11101111        ; 6-0 right turn
             CALL   Z,altervwpl
             SUB    %11111011        ; Q-T move forward
             JR     NZ,rexmove        ; no move, let Rex Move
             CALL   movedir          ; do move in dir
             JR     C,rexmove        ; illegal move, no score
             LD    HL,(rexxy+1)
             SBC    HL,BC
             JR     Z,loop1          ; on rex, no walk for killtest
             LD    HL,score+5        ; each valid move is a point
             CALL   setscore

rexmove      LD    A,0                ; Rex does a move every 2nd
             DEC    A                ; loop
             AND    1
             LD    (rexmove+1),A
             LD    (lastk),A        ; undo movekey too
             JR     Z,loop1
             PUSH   BC                ; save xy player
             LD    BC,(rexxy+1)      ; fetch coordinates of Rex
movenr       LD    A,0                ; Rex does 4 moves
             DEC    A                ; in a direction
             AND    3
             LD    (movenr+1),A
             OR     B                ; +1 move
             OR     C                ; when field is odd on x or y
             AND    1
dirrex       LD    A,0
newdir       JR     NZ,samedir        ; no change of direction
             CALL   rnd              ; change direction
             RRCA                    ; a=a/2, RND doubles
             LD    (dirrex+1),A      ; new direction for rex
samedir      CALL   movedir+2        ; A holds current direction
             JR     C,newdir        ; wall hit, change direction
             LD    (rexxy+1),BC     ; store new rex coordinates
             POP    BC              ; get xy player
loop1        JP     loop            ; back to gameloop

nextnr       LD    (HL),28
setscore     DEC    HL
             INC    (HL)            ; add the score, 1 or 100
             LD    A,(HL)
             CP    38
             JR     Z,nextnr
             RET

delay        LD    HL,frames        ; delay for speed of game
             LD    A,(HL)

```

```

wfr      SUB 20
        CP  (HL)
        JR  NZ,wfr
        RET

wallhit   CALL rnd
        LD  B,A                ; random Y
        CALL rnd
        LD  C,A                ; random X
        CALL field
        JR  C,wallhit         ; which can be a wall
        RET

altervwpl LD  HL,movedir+1     ; alter playerview
        LD  A,(HL)
        ADD A,E
        LD  (HL),A
        RET

rnd
rseed     LD  HL,(frames-1)
        LD  DE,0
        ADD HL,DE
        DEC HL
        LD  A,H
        AND #3F
        LD  H,A
        LD  (rseed+1),HL
        LD  A,R
        ADD A,(HL)
        AND 7                 ; 0-7
        JR  Z,rnd             ; 0 not allowed
        ADD A,A               ; 2,4,6,8,10,12,14
        RET

movedir   ADD  A,0             ; direction you look
        AND  3                 ; when used from other entry
        LD  HL,movetab        ; do the move
        ADD A,L
        LD  L,A
        LD  L,(HL)
        JP  (HL)

; screen per field, w/g,d1,d2,w/g wall or gate
; d1 and d2 must be set on display data1 and data2

hr
synch1    LD  B,14
        DJNZ synch1

        LD  BC,#0A10
        LD  D,empty/256
        LD  HL,datadisp
        LD  A,H
        LD  I,A

nline     PUSH HL              ; linedelay
        POP  HL
        LD  A,(HL)
        LD  A,(HL)
shline    LD  A,(HL)
        CP  rex*256/256-1
        JP  NC,fulludg        ; full udg or compressed
        RET  NC
        JP  compudg           ; space and wall are comp.

```

```

fulludg    LD    A,C                ; index in udg
           ADD    A,A                ; double udg
           ADD    A,(HL)             ; which udgnr
compudg    LD    E,A                ; point to UDG data on line
           INC    L
           INC    L
           LD     A,(DE)             ; fetch udg
           LD     (HL),A            ; set pos2
           INC    E
           INC    L
           LD     A,(DE)             ; fetch udg
           LD     (HL),A            ; set pos3
           DEC    L
           DEC    L
           DEC    L
           LD     A,L
           LD     R,A
           CALL   lbuf+#8000        ; the display of the line

low         DEC    C                ; each maze field has 16 lines
           JR     NZ,nline
           LD     C,16              ; set for next field
           INC    L
           INC    L
           INC    L
           INC    L
           INC    L
           DJNZ   shline

syncrow     LD     B,10
           DJNZ   syncrow

           LD     BC,#307            ; the lowres display
           LD     HL,dfile+#8000
           LD     A,#1E
           LD     I,A
           LD     A,#F5
           CALL   #2B5

exit        CALL   #292
           CALL   #220
           LD     IX,hr
           JP     #2A4

n           EQU    27

dfile       DEFB   #76
score       DEFB   28,28,28,28,28,28,0
hiscore     DEFB   28,28,28,28,28,28,#76
message     DEFW   0,0,0,0,0,0,0
texting     DEFB   #76              ; the messages on screen
           DEFB   "R"-n,"E"-n,"X"-n,0,"I"-n,"S"-n,0
           DEFB   "H"-n,"U"-n,"N"-n,"G"-n,"R"-n,"Y"-n,#76

           DEFB   "R"-n,"E"-n,"X"-n,0,"H"-n,"E"-n,"A"-n,"R"-n
           DEFB   "S"-n,0,"Y"-n,"O"-n,"U"-n,#76
           DEFB   "D"-n,"E"-n,"A"-n,"D"-n,#76
           DEFB   "F"-n,"O"-n,"O"-n,"D"-n,0
           DEFB   "S"-n,"E"-n,"R"-n,"V"-n,"E"-n,"D"-n,#76

space       EQU    #4300-$
           DEFS   space

lbuf        EQU    $

```



```

empty      DEFB 0,0,0                ; linebuffer and empty graphic

          DEFB 0
          RET  NZ
          DEFB 0,0,0

wall       DEFB 255,255              ; wall graphic
rex        DEFB 0,0                  ; rex graphic
          DEFB 56,220
          DEFB 25,152
          DEFB 79,242
          DEFB 127,254
          DEFB 63,252
          DEFB 31,248
          DEFB 54,108
          DEFB 96,6
          DEFB 45,180
          DEFB 127,254
          DEFB 63,252
          DEFB 107,214
          DEFB 99,198
          DEFB 127,254
          DEFB 29,184

hatman     DEFB 0,0                  ; hatman graphic
          DEFB 14,112
          DEFB 2,64
          DEFB 55,236
          DEFB 102,102
          DEFB 55,236
          DEFB 31,248
          DEFB 4,32
          DEFB 9,144
          DEFB 8,16
          DEFB 58,92
          DEFB 56,28
          DEFB 15,240
          DEFB 15,240
          DEFB 15,240
          DEFB 15,240

food       DEFB 0,0                  ; food graphic
          DEFB 7,140
          DEFB 31,242
          DEFB 55,132
          DEFB 51,242
          DEFB 120,140
          DEFB 127,128
          DEFB 127,140
          DEFB 127,242
          DEFB 63,132
          DEFB 31,242
          DEFB 15,140
          DEFB 64,0
          DEFB 100,76
          DEFB 74,170
          DEFB 100,76

          DEFB 0,0

datadisp   DEFS 28                  ; the screendata

basic      DEFB 0,0                  ; only used to start program
          DEFB 0,0
          DEFB 249,212,28
          DEFB 126

```

```

        DEFB 143,0,18,0,0,0      ; 14

init    LD    IX,hr              ; 4
        LD    SP,#4400          ; 3
        JP    start             ; 3

field   LD    HL,maze-3         ; calculate fieldvalue or
        LD    A,B               ; set fieldvalue
        INC   A

fy      INC   HL
        INC   HL
        DEC   A
        JR    NZ,fy

dodx    LD    A,C
        INC   HL
        LD    E,A
        SUB   8
        JR    NC,dodx
        LD    D,127
        LD    A,(HL)

ffnd    RRC    D                ; shift for setting field
        RLCA                ; shift for testing field
        DEC   E
        JR    NZ,ffnd
        RLCA
        RET

maze    EQU    $                ; the space for the maze

vars    DEFB 128

last    EQU    $

```