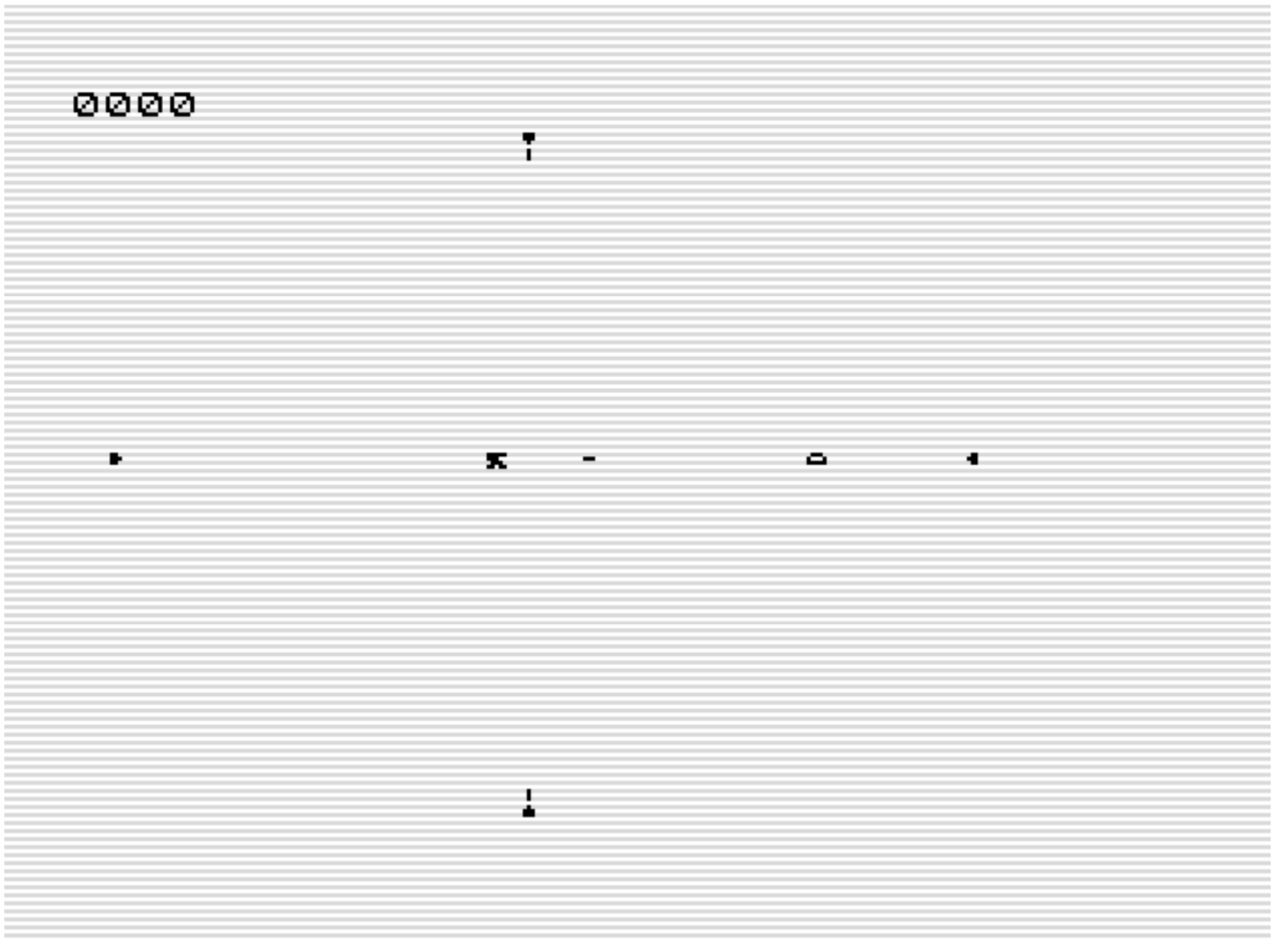


## Gold hunt



**The horizontal movement could be stored in 7 lines at the max. To save memory on the vertical movement the items can never be on the same height. With this trick I can set 1 UDG per line on such a large screen. When the screen was ready the coding was tight to fit it in 1K.**

```
; Gold hunt
```

```
? * TORNADO *
```

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

```
basic LD B,0  
JR init0  
  
DEFB 236,212,28 ; The BASIC  
DEFB 126  
DEFB 143,0,18
```

```
eline DEFW last  
chadd DEFW last-1  
xptr DEFW 0  
stkbot DEFW last  
stkend DEFW last  
berg DEFB 0  
mem DEFW 0 ; not needed without fp  
DEFB 128
```

```
init1 JP init
```

```

lastk      DEFB 255,255,255
margin     DEFB 55
nxtlin     DEFW basic

init0      XOR   A                      ; also for 48K bug only
           DEFB 254
flagx      DEFB 0

yxmove     JR    init1

taddr      DEFW 3213                    ; used by ZX
           DEFB %11110111
           DEFB %11101111

frames     DEFW 65535
coords     DEFB 0,0
prcc       DEFB 188
sposn      DEFB 33,24
cdflag     DEFB 64

canupos    DEFB 174                    ; ypos canup highest cannon
           DEFB 15                      ; xpos canup
           DEFB canup*256/256          ; UDG-pointer, show cannon

goldpos    DEFB 78
           DEFB 5
           DEFB gold*256/256

candwnpos  DEFB 6
           DEFB 14
           DEFB candown*256/256

fiuppos    DEFB 162
           DEFB 15
           DEFB firev*256/256

fidopos    DEFB 10
           DEFB 15
           DEFB firev*256/256

filepos    DEFB 0
           DEFB 3
           DEFB fireh*256/256

firepos    DEFB 0
           DEFB 25
           DEFB fireh*256/256

; set ypos on restart
canlepos   DEFB 0
           DEFB 1
           DEFB canleft*256/256

canrepos   DEFB 0
           DEFB 28
           DEFB canright*256/256

playpos    DEFB 16*8
           DEFB 11
           DEFB player*256/256

items      DEFB 101                    ; The top of

```

```

        DEFB 12                ; letter G on startscreen
        DEFB %00011100

; this memory is used after loading
; for display of items. After loading
; used to set up the program

init      LD    IX,hr          ; Hires mode
          LD    SP,#4300       ; SP in game, not at end
          EX    AF,AF'        ; delay intrupt

          LD    HL,canleft     ; linebuffer over sysvar
          LD    DE,#4000
          LD    C,31          ; this copy needed
          LDIR                ; before 48K repair

          LD    L,C            ; 0
          LD    H,D            ; HL = #4000
          LD    E,L
          LD    D,#C0          ; DE = #c000
          LD    B,4            ; BC = #400
          LDIR                ; repair 48K bug

          LD    HL,l43f9       ; unloadable in 1k
          LD    DE,#43F9       ; so copy after loading
          LD    C,7            ; to right addresses
          LDIR

cls        LD    HL,#431F      ; clear the items line
          DEC    L
          LD    (HL),C
          JR     NZ,cls

          JP     defkey         ; do redef keys

; move to not loadable locations in 1K
L43f9      LD    BC,(lastk)     ; this in screen area
          LD    A,C            ; saves 4 bytes in the code
          INC    A
          RET

keydisp    DEFB "D"-27         ; UP, DOWN, LEFT, RIGHT
          DEFB "L"-27         ; display table
          DEFB "R"-27

; max 6 items off the main screen
hr         LD    HL,lowres+#8000 ; the lowres display
          LD    BC,#209
          LD    A,#1E
          LD    I,A
          LD    A,#FB
          CALL #2B5

          LD    B,7            ; sync hires display
hr00       DJNZ hr00

          LD    HL,screens     ; the screen with player
          LD    A,H
          LD    D,H
          LD    I,A           ; set highy byte of hr-screen
          EXX
          LD    HL,items       ; the vertical items
          LD    BC,#B140       ; 176 lines +1

```

```

low          DEC    B                ; on RET here NEVER to zero

lineloop     LD     A,B              ; check item screen
             CP     (HL)
             JR     NZ,testscr       ; display loose items

doitems      INC     L
             LD     E,(HL)           ; fetch position of item
             INC    HL
             LDI                    ; copy item to LBUF
             LD     C,E              ; save "position" for erase
             LD     A,0              ; XOR A is too fast
             LD     R,A              ; set lowbyte
             JP     #C000            ; display the item

testscr      EXX
             SUB    (HL)             ; check screen
             JR     NZ,doempty       ; display screen

doscr        INC     HL
             LD     A,L
             LD     L,(HL)           ; "calculate" next line
             EXX
             LD     R,A
             JP     #C000            ; show screenline

doempty      LD     A,7
delay        DEC     A               ; timefill an empty line
             JR     NZ,delay

             EXX                     ; back to maintests

             LD     E,C              ; erase last printed item
             DEC    DE               ; from LDI 1 too far
             LD     (DE),A           ; erase for next item
             LD     (DE),A           ; timing
             INC    DE               ; timing

             DJNZ   lineloop         ; LAST LINE EMPTY, only exit

             CALL   #292             ; back from intrupt
             CALL   #220
             LD     IX,hr
             JP     #2A4

linescr      POP     DE              ; get UDGpointer
             LD     C,A              ; save counter
             LD     A,L
             ADD    A,B              ; get x position
             LD     L,A
             INC    HL
             LD     A,(DE)
             AND    126              ; take of bit 0
             LD     (HL),A
             POP    HL
             LD     A,C              ; Y pointer back
             JR     linein

makescreen   LD     HL,canlepos
             LD     DE,playpos
             LD     A,(DE)
             CP     (HL)
             LD     B,(HL)           ; get y cannonleft
             LD     C,3

```

```

        JR    C,ok1                ; do cannonscreen
        LD    B,A                  ; get y player
        EX    DE,HL               ; swap for later
        INC   C                   ; we need 3 or 4, so do 4
ok1     PUSH  AF                  ; save flags
        PUSH  BC                  ; save nr lines
        PUSH  DE                  ; save next Y position
        LD    HL,scrlines
        CALL  stfnd               ; make lines
        POP   DE
        POP   BC
        LD    A,7
        SUB   C
        LD    C,A                ; compute complement to do
        LD    A,(DE)
        LD    B,A                ; get next y
        POP   AF                 ; get flags
        CALL  NZ,stfnd           ; add lines when needed

        LD    DE,items           ; bubblesort
        LD    C,10               ; all items to print
fnext   LD    B,10               ; so display goes ok
        XOR   A                  ; reset highest
        LD    HL,canupos        ; point to itemrecords
fihigh  CP    (HL)               ; check higher
        INC   HL
        JR    NC,nothi          ; lower than current highest
        BIT   7,(HL)
        JR    NZ,nothi          ; already found earlier
        LD    A,L
        DEC   HL
        LD    (nfound+1),A      ; pointer current highest
        LD    A,(HL)            ; get current highest
        INC   HL
nothi   INC   HL
        INC   HL
        DJNZ  fihigh            ; check all items

        PUSH  HL
        PUSH  BC

nfound  LD    L,0                ; point to x-pos in record
        LD    B,(HL)            ; get x-pos
        SET   7,(HL)            ; signal highest found
        INC   HL
        LD    L,(HL)            ; get udg pointer

; here A holds linenumber, first test on screen
; otherwise write to items
doudg   PUSH  DE                ; save for after scrtest
        PUSH  HL                ; when write to items

seeknext LD    HL,scrlines-30
        LD    DE,30
        ADD   HL,DE
        LD    D,(HL)
        CP    (HL)
        JR    Z,linescr        ; display on "screen"
        JR    C,seeknext
        POP   DE                ; get stacked pairs
        POP   HL                ; but also ex de,hl

; not on screen so do on items
        LD    (HL),A            ; items display is

```

```

        INC HL ; done by hr-routine
        LD (HL),B
        INC HL
        LD C,A ; save ypointer

        LD A,(DE) ; get udg
        AND 126 ; take of bit 0
        LD (HL),A

        LD A,C
        INC HL
        LD (HL),A ; stop show at the end

linein  EX DE,HL
        DEC A
        BIT 0,(HL) ; check endmarker UDG
        INC HL
        JR Z,doudg

        POP BC
        POP HL
        DEC C
        JR NZ,fnext

        LD B,10 ; all items need reset
        LD HL,canupos+1 ; of check bit
resx    RES 7,(HL)
        INC HL
        INC HL
        INC HL
        DJNZ resx

testhit LD B,5
        LD HL,fiuppos ; test all fires
fihit  DEC B
        RET Z
        LD DE,playpos
        LD A,(DE) ; get y player
        XOR (HL) ; compare y fire
        AND #F8 ; take of <8
        INC HL
        INC DE
        JR NZ,hlinc ; not same y coordinate
        LD A,(DE) ; get x player
        XOR (HL) ; check x fire
hlinc  INC HL
        INC HL
        JR NZ,fihit ; check next fire
        POP HL ; drop call ret

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

        LD H,A
        LD L,A
        LD (yxmove),HL ; reset bullit movement
        LD HL,#1C1C ; "00"
        LD (score),HL
        DEC H ; "0(-1)"
        LD (score+2),HL ; -1 corrected by score
        LD A,86 ; mod 8 = 6
        LD (canlepos),A
        LD HL,playpos

```

```

        LD    (HL),94            ; player y
        LD    A,14
        LD    (canupos+1),A
        INC   HL
        LD    (HL),15           ; player x

nextgold LD    HL,score+4
        DEFB  17                ; hide ten
ten      LD    (HL),28
        DEC   HL
        INC   (HL)
        LD    A,(HL)
        CP    38
        JR    Z,ten

goldrnd  LD    B,25
        CALL  rnd
        LD    (goldpos+1),A      ; random x for gold

        LD    B,19
        CALL  rnd
        ADD   A,A
        ADD   A,A
        ADD   A,A
        DEC   A
        DEC   A
        LD    (goldpos),A        ; random y for gold

gameloop LD    DE,(goldpos)
        LD    HL,(playpos)
        SBC   HL,DE              ; never carry here
        JR    Z,nextgold        ; gold found

        LD    HL,frames          ; delay to make
        LD    A,(HL)             ; the hard but playable
        SUB   5
wfr      CP    (HL)
        JR    NZ,wfr

; move horizontal bullits
xbullit LD    DE,yxmove
        LD    HL,playpos

        LD    A,(DE)             ; get x move
        OR    A
        LD    BC,canlepos
        JR    NZ,movexbul        ; fired, move bullit

        LD    A,(BC)             ; get y cannon left
        CP    (HL)
        JR    NZ,movecanx        ; when same fire
        LD    (DE),A            ; signal x movement

movexbul LD    A,(firepos+1)      ; move right bullit
        DEC   A
        LD    (firepos+1),A

        LD    A,(filepos+1)      ; move left bullit
        INC   A
        LD    (filepos+1),A
        SUB   15
        JR    NZ,ybullit        ; check hit eachother

        LD    (DE),A            ; reset movement

```

```

movecanx    LD    A,2                ; reset fire
            LD    (filepos+1),A      ; bullits back to cannon
            LD    A,27
            LD    (firepos+1),A
            LD    A,(BC)             ; get y cannon
            CP    (HL)               ; check position player
            JR    C,movecanup        ; do right movement towards
            SUB    16                 ; move down
movecanup   ADD    A,8                ; move up
            LD    (BC),A             ; save new positions
            LD    (canrepos),A
            DEC    A                  ; save new fires
            LD    (filepos),A
            LD    (firepos),A

ybullit     INC    DE                ; same routine for
            INC    HL                ; horizontal cannons
            LD    C,canupos*256/256+1
            LD    A,(DE)
            OR     A
            JR    NZ,moveybul

            LD    A,(BC)
            CP    (HL)
            JR    NZ,movecany

            LD    (DE),A             ; signal y movement

moveybul     LD    A,(fiuppos)
            SUB    8
            LD    (fiuppos),A
            LD    A,(fidopos)
            ADD    A,8
            LD    (fidopos),A
            SUB    98
            JR    NZ,movedone
            LD    (DE),A

movecany     LD    A,170
            LD    (fiuppos),A
            LD    A,10
            LD    (fidopos),A
            LD    A,(BC)
            CP    (HL)
            JR    C,movecanre
            DEC    A
            DEC    A
movecanre    INC    A
            LD    (BC),A
            LD    (candwnpos+1),A
            LD    (fiuppos+1),A
            LD    (fidopos+1),A

movedone     CALL  makescreen        ; cannonmove done, make screen

playmove     LD    A,0
            DEC    A
            AND    7
            LD    (playmove+1),A
            JR    Z,false            ; no movement 1/8

            CALL  #43F9              ; get last key pressed
            CALL  NZ,#7BD            ; translate when key down

```



```

LD    HL,keydest*256/256+#4000
LD    DE,playpos
LD    BC,#08A6            ; up
CP    (HL)
INC   HL
JR    Z,domove
LD    BC,#F80D            ; down
CP    (HL)
INC   HL
JR    Z,domove
CP    (HL)
INC   HL
INC   DE
LD    BC,#FF02            ; left
JR    Z,domove
CP    (HL)
LD    BC,#011B            ; right
JR    NZ,false

domove LD    A,(DE)            ; get x or y
CP    C                    ; test against max
JR    Z,false
ADD   A,B                  ; do move
LD    (DE),A               ; save new position

false  CALL makescreen      ; player moved built screen
JP     gameloop

stfnd  LD    DE,30           ; find start of line
LD    (HL),B               ; save y
DEC   B                    ; next line is 1 less
LD    A,L
ADD   A,E
INC   HL
LD    (HL),A               ; set next pointer
DEC   E
fi     INC   HL
LD    (HL),D               ; clearlines
DEC   E
JR    NZ,fi
DEC   C
JR    NZ,stfnd
RET

rnd     LD    DE,0           ; standard rnd routine
LD    HL,(frames)
ADD   HL,DE
DEC   HL
LD    A,H
AND   #1F
LD    H,A
LD    (rnd+1),HL
LD    A,(HL)

frnd   SUB   B
JR    NC,frnd
ADC   A,B
INC   A
INC   A
RET

space  EQU   #4300-$
DEFS  space

```

```

; UDG's are hidden in the linebuffer
; which is copied over the sysvar
; this memory is then cleared and used as
; linedata for items to display.

canleft    DEFB %00001100
           DEFB %00001110
           DEFB %00001101

canup      DEFB %00011100          ; name udg defines position
           DEFB %00011100          ; not how it looks
           DEFB %00001001

gold       DEFB %00011100
           DEFB %00100010
           DEFB %00111111

player     DEFB %00111110
           DEFB %00101010
           DEFB %00011100
           DEFB %00110111

firev      DEFB %00001000
           DEFB %00001000
           DEFB %00001001

fireh      DEFB %00111001

canright   DEFB %00011000
           DEFB %00111000
           DEFB %00011001

candown    DEFB %00001000
           DEFB %00011100
           DEFB %00011101

keydest    DEFB 0                  ; marker for #40.. only
           DEFB 0
           DEFB 0
           DEFB 0,0
           JP low                  ; only needed for copy

; scorescreen fills the gap needed here.
           DEFB 0
lowres      DEFB 118
score      DEFB "U"-27,0,0,0
           DEFB 118

; below is the hires screen, during init
; it shows a flashing screen
; after starting the screen is built here
scrlines    DEFB 100              ; a nice demo on start
           DEFB demo2*256/256

           DEFB 0,60,0,60,0,60,0,60,0,60,0
           DEFB 32,228,112,37,46,112
           DEFB 0,60,0,60,0,60,0,60,0,60,0

demo2       DEFB 99
           DEFB demo3*256/256

           DEFB 0,66,0,66,0,66,0,66,0,66,0
           DEFB 37,20,72,61,41,32
           DEFB 0,66,0,66,0,66,0,66,0,66,0

```

```

demo3      DEFB 98
           DEFB demo3*256/256

           DEFB 0,126,0,126,0,126,0,126,0,126,0
           DEFB 60,231,112,37,233,32
           DEFB 0,126,0,126,0,126,0,126,0,126,0

defkey     LD HL,keydisp-1
           LD DE,keydest*256/256+#4000
redef      LD A,(lastk)          ; DE = marker in #40..
           INC A
           JR NZ,redef
           PUSH HL
waitkey    LD HL,scrlines        ; flash the loading screen
           LD A,H
           SUB (HL)
           LD (HL),A
           INC A
           LD (items),A
           LD HL,frames
           LD A,(HL)
           SUB 20
flashwait  CP (HL)
           JR NZ,flashwait
           LD BC,(lastk)
           LD A,C
           INC A
           JR Z,waitkey
           PUSH DE
           CALL #7BD             ; translate key
           POP DE
           POP HL
           LD (DE),A
           INC DE
           INC HL
           LD A,(HL)
           LD (score),A
           SUB 33                ; LD HL from HR
           JR NZ,redef
           LD (score),A         ; clear final direction
           JP start             ; init done, goto gamestart

vars      DEFB 128
?
last      EQU $

```