

## Mission Impossible: Bombsquare



**This game was on my list. All it needed was the right displayroutine. Goat and wolves has that routine. The routine is used 1:1 for this game. It only needed the graphics from this game. A lot from game 67 could be reused. So the actual coding of the game was done in 2 days.**

```
; Mission Impossible
; Game 69 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 #8A,192,0,#9A-1  ; show bomb table
           DEFB #9A,124,28,#AA-1
           DEFB #AA,54,42,#BA-1
           DEFB #BA,115,62,#CA-1
           DEFB #CA,127,54,#DA-1
           DEFB #DA,127,93,#9A-1
           DEFB #9A,62,28,#CA-1
           DEFB #CA,28,54

grtab2     DEFB #9A,28,28,#9A-1  ; no bomb table
           DEFB #9A,28,28,#AA-1
           DEFB #9A,28,28,#BA-1
           DEFB #9A,28,28,#CA-1
           DEFB #9A,28,28,#DA-1
           DEFB #9A,28,28,#9A-1
           DEFB #9A,28,28,#CA-1
           DEFB #9A,28,28

; 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

; people screen must be between ..80 to ..ff
people     DEFW 0,0,0,0,0,0,0,0

p1          DEFB 28,28,28,28,28,28,28,28
p2          DEFB 28,28,28,28,28,28,28,28
p3          DEFB 42,42,42,42,42,42,42,42
p4          DEFB 62,62,62,62,62,62,62,62
p5          DEFB 54,54,54,54,54,54,54,54
           DEFB 93,93,93,93,93,93,93,93

```

```

DEFB 93,93,93,93,93,93,93,93

; 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
        NOP

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

        LD    HL,retlbuf        ; return from highmem
        EXX
        LD    D,A                ; copy bomb over people

        LD    H,graphtab/256    ; preset highbyte bombudg
        LD    (saveesp+1),SP
        LD    SP,#4000          ; displaystack is on sysvar
yposplay LD    B,0                ; 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
nrdead  DEFB 28,28,0
        DEFB "M"+x,"I"+x,142,"B"+x,"O"+x
        DEFB "M"+x,"B"+x,128,"A"+x,"T"+x,128
coord   DEFB 156,156,0
sumdead DEFB 28,28,0
hiscore DEFB 28,28,34,37        ; "69"
        DEFB 118

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

bloop   LD    L,graphtab*256/256 ; default,we copy the bomb
        DEC    B                ; next line to show
        LD    C,31              ; 8 lines 16 copies 7 fillers
        LD    A,#8A-1          ; first line to show
        CP     B                ; test against show bomb
        JR    NZ,setl          ; bomb on other line
        JR    cloop            ; for same timing

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

cloop   LD    E,(HL)            ; get x bomb
        NOP                    ; filler
        INC    HL                ; point to data bomb/no bomb
        LDI    ; copy bomb/no bomb
        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

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

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

        CALL  shsumdead

nextbomb LD    A,4
        LD    (lastk-1),A

        CALL  rnd
        CALL  rnd
        LD    C,A
        LD    (endtest+1),BC  ; save bomb position

newround LD    HL,lastk-1
        DEC   (HL)
        JR    NZ,cls          ; 3 rounds played

        LD    E,64            ; display off
        CALL  evacuate        ; remove people from square

endtest  LD    BC,0
        LD    E,128           ; show casualties
        CALL  evacuate        ; now this is the kill
        PUSH  AF
        CALL  shownr          ; show casualties
        CALL  showbomb        ; show bombposition
        POP   AF
        LD    B,A
totdead  ADD    A,0
        LD    (totdead+1),A
        CP    25
        CALL  C,shsumdead
        JR    NC,eog
        LD    A,50
        SUB   B
        LD    B,A

addpoint LD    HL,score+4
        DEFB 17

```

```

ten      LD      (HL),28
        DEC     HL
        INC     (HL)
        LD      A,(HL)
        CP      38
        JR      Z,ten
        DJNZ    addpoint

        LD      A,H                ; delay to show
        LD      HL,frames          ; where bomb actual was
        ADD     A,(HL)
wfr      CP      (HL)
        JR      NZ,wfr

        JR      nextbomb

cls      LD      HL,lbuf1
clscr    LD      B,16
ccline   LD      (HL),#64          ; display off
        INC     HL
        DJNZ    ccline
        INC     L                  ; skip jp hl
        JR      NZ,clscr          ; on #4400 done

nrpeop   LD      B,50              ; display nr people
makepeople PUSH   BC              ; save counter
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 people shown
        JR      Z,nxtrnd         ; not on used field
        LD      (HL),B           ; set next man to show
        POP     BC

        DJNZ    makepeople        ; set all people

; calculate and show casualties
        LD      BC,(endtest+1)
        CALL    dbb              ; show once

        LD      BC,0

playloop LD      HL,lastk-1        ; stepcounter over sysvar
        CALL    showbomb          ; show bomb as cursor

        PUSH    BC              ; save xy for after keypress

        LD      HL,coord
        LD      A,B
        ADD     A,38+128          ; show Y coordinate inverted
        LD      (HL),A
        INC     HL
        LD      A,C
        ADD     A,38+128          ; show X coordinate inverted
        LD      (HL),A

        CALL    field            ; preload HL to clear display

wup      INC     A                ; test nokey pressed

```

```

wdown      LD    BC,(lastk)      ; get lastkey
           LD    A,C             ; port to A, no key = 255
           JR    NZ,wup
           INC   A               ; test key pressed
           JR    Z,wdown

setold      LD    (HL),#64        ; erase bomb

           CALL  #7BD            ; get keycode
           POP   BC
           DEC   A
           JP    Z,newround      ; Z goto next round
           PUSH  BC              ; save old xy
           CP    10-1            ; Q pressed?
           JR    NZ,t2
           DEC   B
t2          CP    5-1             ; A pressed?
           JR    NZ,t3
           INC   B
t3          SUB   26-1            ; O pressed?
           JR    NZ,t4
           DEC   C
t4          INC   A               ; P pressed?
           JR    NZ,testmove
           INC   C
testmove    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
           DEFB  62

false       POP   BC             ; get unchanged XY player

valid       CALL  dbb            ; count deaths on this field
           JP    playloop

; this routine can evacuate and call numbers of victims
evacuate    LD    H,7            ; bomb goes 3 up 3 down
           XOR    A              ; casualtycounter
ybomb       LD    L,7            ; bomb goes 3 left 3 down
xbomb       PUSH  HL            ; save radius
           PUSH  BC              ; save bomb xy
           PUSH  AF              ; save nr dead
           LD    A,B             ; Y of bomb
           ADD   A,H
           SUB   4               ; radius now in Y
           LD    B,A             ; set field reached
           LD    A,C
           ADD   A,L
           SUB   4
           LD    C,A             ; same with X
           PUSH  DE              ; save evacuate/show
           CALL  field           ; get field reached
           POP   DE              ; get pointer
           POP   AF              ; get nr
           BIT   6,(HL)          ; test visible
           JR    NZ,noman
           LD    (HL),E          ; show or invert
cntdead     INC   A              ; count nr people
noman       POP   BC             ; get original xy
           POP   HL              ; get radius
           DEC   L

```

```

        JR    NZ,xbomb            ; 3 left bomb 3 right
        DEC   H
        JR    NZ,ybomb            ; 3 up bomb 3 right
        RET

dbb      LD    E,0                ; indicator normal
        CALL  evacuate            ; count only

shownr   LD    HL,nrdead          ; A to text
        LD    (HL),27
setten   INC    (HL)
        SUB   10
        JR    NC,setten          ; make 10 digit visible
        INC   HL
        ADD   A,38                ; set unit
        LD    (HL),A
        RET

shsumdead LD    (totdead+1),A      ; write new nr dead
        LD    HL,sumdead          ; now nr to text
        JR    shownr+3

showbomb CALL  field
        LD    A,(HL)              ; get old background
        LD    (setold+1),A        ; save for later
        LD    (HL),B              ; show field player

        LD    HL,graphtab
        LD    A,(HL)              ; get old pointer+x
        SUB   #8A                 ; take off first pointer
        LD    E,A                 ; old x to e
        LD    D,8                 ; 8 lines to write pointers
setbomb  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,setbomb

        LD    A,B                 ; get Y bomb
        ADD   A,#8A               ; add pointer
        LD    (yposplay+1),A      ; set bombdisplay on
        RET

size     EQU   16*17

st       EQU   26                  ; stack size

; some space left, is also needed for the stack
space    EQU   #4400-size-st-22-$

        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

field    LD    A,15
        LD    HL,lbuf1-17         ; (HL) has bit 6 set, errorfld
        CP    B

```

```

        RET    C                ; out of board Y direction
        CP     C
        RET    C                ; out of board X direction

        PUSH   BC
        INC    B
        LD     DE,17
fline   ADD     HL,DE            ; find right lbuf
        DJNZ   fline
        ADD     HL,BC          ; find position in lbuf
        POP    BC
        RET

lbuf1    DEFW   0,0            ; on load all fields invisible
        DEFW   0,0            ; program will determine
        DEFW   0,0            ; which fields need to show
        DEFW   0,0            ; the UDG on that field
errorfld 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 dsisplaystack from screen
        LD     DE,#4000        ; destination: sysvar
        LD     C,34            ; copy stack to now free mem
        LDIR                    ; save 34 bytes

        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,field        ; 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  $
```