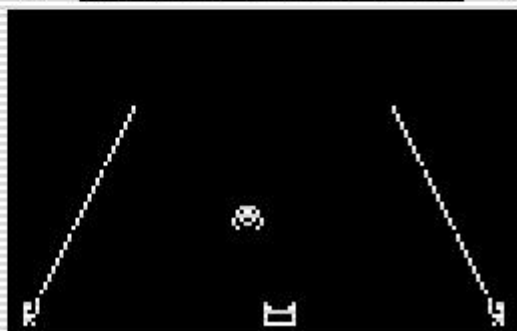


Ghostbusters

0000 3 GHOSTBUSTERS 84 0060



Who you gonn call..... GHOSTBUSTERS! The epic ghostcatching part from Ghostbusters is now on the ZX81. The display from Startrip is base for the 3 graphics on a line.

```
; GHOSTBUSTERS
; Game 60 in 1K hires for the ZX81
```

```
? * TORNADO *
```

```
sr      EQU 32                ; size of stack during game
hrstack EQU #4400-110

        ORG #4009
        DUMP 49161

basic   LD D,#C0                ; preset for 48K bug
        JR init0                ; this game has no 48K bug

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

eline   DEFW last                ; needed to load
chadd   DEFW last-1
xptr    DEFW 255
stkbot   DEFW last                ; needed to load
stkend   DEFW last                ; needed to load
berg     DEFB 0
mem      DEFW 0
        DEFB 000
```

```

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
nxtlin     DEFW basic                ; reusable after load

init0      LD     E,L                ; delay intrupts by
           DEFB  #26                 ; LD H,64
flagx      DEFB 64                   ; clever setting of flags

           XOR    A                  ; intruptcounter reset
           EX     AF,AF'

taddr      DEFW 0                    ; used by ZX81,no hurting code
           LD     B,4                ; frames is set ok

frames     DEFW #DD*256+1            ; used by ZX81, clever IX set
coprcc     LD     HL,hr              ; set IX
sposn      JR     init1
cdflag     DEFB 64                   ; used by zx81

ghostudg   DEFB 10,60
           DEFB 10,90
           DEFB 10,126
           DEFB 10,165
           DEFB 10,153
           DEFB 10,66
           DEFB 10,0

           DEFB 0,80,48,112,144,176,176
udgmr1     DEFB 128

           DEFB 0,3,3,7,9,11,11
udgmr2     DEFB 8

           DEFB 0,128,64,64,32,32,16
udgfl1     DEFB 16

           DEFB 8,8,4,4,2,2,1
udgfl2     DEFB 1

           DEFB 78,114,66,78,114,66,78
udgsp      DEFB 114

lbuf       LD     R,A                ; display 16 columns
noudg      DEFB #80,#80,#80,#80      ; displaybuffer and
           DEFB #80,#80,#80,#80      ; captured ghost (no show) udg
           DEFB #80,#80,#80,#80
           DEFB #80,#80,#80,#80

HIER       JP     Z,bloop            ; 48K bug
           JP     NC,cloop           ; 48K bug
           JP     savesp             ; exit with extra blanc line

           DEFB 16,16,32,32,64,64,128
udgfr1     DEFB 128

           DEFB 0,1,2,2,4,4,8
udgfr2     DEFB 8

;udgtrc    DEFB 12,0,12,0

```

```

;      DEFB 12,255,12,231,12,195
;      DEFB 12,255,12,0

udgtro  DEFB 12,129,12,195
        DEFB 12,255,12,129,12,129
        DEFB 12,255,12,0

;      DEFB 66,66,129,129,153,165,195
;udgcop      DEFB 129

;      DEFB 66,66,129,129,153,153,189
;udgccl      DEFB 66

energy  LD    A,0
        DEC   A
        RET   Z

fullen  LD    (energy+1),A
        SRL   A
        LD    HL,power
        LD    (HL),27

sett    INC    (HL)
        SUB   10
        JR    NC,sett
        ADD   A,38
        INC   HL
        LD    (HL),A
        RET

eog     LD    HL,score-1
        LD    DE,hiscore-1
        LD    BC,5

fihi    INC    HL
        INC    DE
        DEC    C
        LD    A,(DE)          ; when C=0 (DE)=118
        CP    (HL)           ; and (HL)=0
        JR    Z,fihi         ; so NOT equal and no
        CALL  C,#19F9        ; hiscore with same score

; this 8 bytes can be skipped when needed
loadstart LD    HL,#4012
cldisp    DEC    L
        LD    (HL),B
        JR    NZ,cldisp

start    LD    A,191          ; game over, wait for
        IN    A,(254)
        RRA                ; newline
        JR    C,start

        LD    HL,#1C1C
        LD    (score),HL
        LD    (score+2),HL

        LD    A,32
        LD    (lives),A

dead     LD    HL,lives
        DEC    (HL)
        LD    A,(HL)
        CP    28
        JR    Z,eog

```

```

nround      LD    A,199
            CALL  fullen

            LD    A,3
            LD    (menl+1),A
            LD    A,R
            AND    7
            LD    H,5
            ADD    A,H
            LD    L,A
            LD    (ghostxy+1),HL

            LD    A,32
            LD    (menr+1),A

            LD    A,22
            LD    (gmove),A

            XOR    A
            LD    (beam+1),A          ; reset beamshot

            LD    A,200
            CALL  delay
            LD    HL,hrstack+4        ; erase old display on stack

cls          LD    (HL),noudg*256/256
            INC    HL
            LD    (HL),noudg/256
            INC    HL
            XOR    A
            LD    (HL),A              ; B=0 from hiscore or init
            INC    HL
            LD    (HL),A
            LD    A,7
            ADD    A,L
            LD    L,A
            JR     NC,cls              ; clear the whole displaystack

            CALL  energy

menl         LD    BC,#A03
            CALL  prudg12-7

menr         LD    BC,#A20
            LD    DE,udgmr1*256/256+#400
            LD    HL,hrstack-8
            CALL  prudg12

ghostxy     LD    DE,ghostudg
            LD    BC,#307
            CALL  udg3xyset

            LD    DE,udgtro
            CALL  udg3xyset-3

beam        LD    A,0
            DEC    A
            JR     NZ,beam2

            LD    HL,(menl+1)
            LD    DE,(menr+1)
            PUSH   HL

```

```

SBC HL,DE
LD A,L
POP HL
CP 243
JP NC,dead
LD A,6
sbeam PUSH AF
DEC H
INC L
DEC D
DEC E
PUSH HL
PUSH DE
LD B,H
LD C,L
LD E,udgfl1*256/256
CALL prudg12-5
POP BC
PUSH BC
LD DE,udgfr1*256/256+#400
LD HL,hrstack-8
CALL prudg12
POP DE
POP HL
POP AF
DEC A
JR NZ,sbeam
JR input

beam2 INC A
JR Z,input
beamy LD BC,#912

updown LD A,255
ADD A,B
LD B,A
LD (beamy+2),A
PUSH AF

prbeam LD E,udgsp*256/256
CALL prudg12-5
INC B
LD A,B
SUB 10
JR C,prbeam

POP AF
SUB 10
LD HL,updown+1
CP 249
JR NC,input

XOR A
SUB (HL)
LD (HL),A
INC A
JR NZ,input

LD BC,(men1+1) ; clear print at bottom
CALL prudg12-7

LD A,(gmove)
CP 24
JP NZ,dead

```

	LD	A, (energy+1)
	SRL	A
	JR	Z, nosc
	LD	B, A
addsc	LD	HL, score+4
	DEFB	17
ten	LD	(HL), 28
	DEC	HL
	INC	(HL)
	LD	A, (HL)
	CP	38
	JR	Z, ten
	DJNZ	addsc
nosc	LD	HL, text
	LD	B, 12
gbflash	RES	7, (HL)
	PUSH	HL
	LD	A, 252
	CALL	delay
	POP	HL
	SET	7, (HL)
	INC	HL
	DJNZ	gbflash
	JP	nround
input	LD	BC, (lastk)
	LD	A, C
	INC	A
	CALL	NZ, #7BD
	CP	1
	JR	NZ, inplr
oldfire	CP	0
	JR	Z, inplr
	LD	HL, beam+1
	INC	(HL)
inplr	LD	(oldfire+1), A
	CP	25
	JR	NZ, tleft
	LD	A, (menl+1)
	CP	27
	JR	NC, errmove
	INC	A
	LD	(menl+1), A
	JR	errmove
tleft	CP	26
	JR	NZ, errmove
	LD	A, (menr+1)
	CP	8
	JR	Z, errmove
	DEC	A
	LD	(menr+1), A
errmove	LD	HL, (frames)
rseed	LD	DE, 0
	ADD	HL, DE
	DEC	HL
	LD	A, H
	AND	#1F
	LD	H, A
	LD	A, (HL)

```

        AND 3
        LD (rseed+1),HL

gmove   LD HL,(ghostxy+1)
        JR skip
        JR NZ,noup
        DEC H
        JR Z,errmove
noup     DEC A
        JR NZ,nodown
        INC H
nodown   DEC A
        JR NZ,noleft
        DEC L
        JR Z,errmove
noleft   DEC A
        JR NZ,chmove
        INC L
chmove   LD A,L
        CP 17
        JR NC,errmove
        LD A,H
        CP 10
        JR NC,errmove

        PUSH HL
        LD B,H
        LD C,L
        LD HL,hrstack-4
fhr      LD A,L
        ADD A,10
        LD L,A
        DJNZ fhr
        LD A,C
        CP (HL)
        INC HL
        JR Z,captest
        CP (HL)
        JR NZ,moveok
captest  INC HL ; test in capturebeam
        LD A,(HL)
        SUB udgsp*256/256
        POP HL
        JR NZ,errmove
        LD C,A
        LD B,H ; out of screen
        LD (ghostxy+1),HL
        LD DE,ghostudg
        PUSH BC
        CALL udg3xyset
        POP HL
        LD (ghostxy+1),HL
        LD A,24
        LD (gmove),A
        DEFB 62

moveok   POP HL

        LD A,(energy+1)
        RRA
        JR NC,skip
        LD (ghostxy+1),HL
skip     LD A,255-9

```

```

        JP      cl

delay   LD      HL,frames
        ADD     A,(HL)
wfr     CP      (HL)
        JR      NZ,wfr
        RET

        LD      E,udg1*256/256
        LD      HL,hrstack-2
        LD      D,#FF
prudgl2 PUSH    BC
        LD      A,C
        SRL     A
        LD      C,A
        JR      NC,fhl
        LD      A,E
        ADD     A,8
        LD      E,A
fhl     LD      A,10
        ADD     A,L
        LD      L,A
        DJNZ    fhl
        LD      (HL),E
        ADD     A,D
        LD      L,A
        LD      (HL),C
        POP     BC
        RET

        LD      BC,#A09
udg3xyset LD     HL,hrstack+4-10
        PUSH    BC
fhly    LD      A,L
        ADD     A,10
        LD      L,A
        DJNZ    fhly
        LD      (HL),E
        INC     HL
        LD      (HL),D
        EX      DE,HL
        LD      B,7
fhlx    LD      (HL),C
        INC     HL
        INC     HL
        DJNZ    fhlx
        POP     BC
        RET

hr      LD      HL,lowres+#8000      ; the lowres display
        LD      BC,#0269            ; minimum lines in this game
        LD      A,#1E                ; needed to prevent scrolling
        LD      I,A
        LD      A,#FB
        CALL    #2B5

        LD      IY,lbuf+#8000        ; JP (IY) faster than JP NN

        EXX                          ; program uses shadowregs
        PUSH    BC                    ; must be saved too
        PUSH    DE
        PUSH    HL

```



```

hr00      LD    B,16                ; outline delay for hires
          DJNZ  hr00

          LD    (savesp+1),SP      ; save current stack
          LD    SP,hrstack        ; use display stack
          LD    A,#40
          LD    I,A
          LD    D,A                ; dataline on #40..
          EXX
          LD    D,A                ; dataline on #40..

bloop     DEFB  #DD
          LD    L,7                ; LD IX,7
          POP   AF                ; get return flag
          POP   BC                ; BC holds UDG2
          POP   HL                ; HL holds xpos and udg3 rev
          EXX
          POP   BC                ; positions UDG1 and UDG2
          POP   HL                ; HL holds UDG1

nline     LD    A,(HL)            ; get data udg1
          LD    E,B                ; set pos udg1
          LD    (DE),A            ; set udg1 on pos1
          LD    A,C                ; get pos2
          EXX
          LD    E,A                ; set pos udg2
          LD    A,(BC)            ; get data udg2
          LD    (DE),A            ; set udg2 on pos2

          LD    A,1                ; set display start
          JP    (IY)              ; do hires display

; mainudg in reverse in memory
; udg3 xpos data increase

cloop     LD    E,(HL)            ; get xpos udg3
          INC   L                  ; point to data udg3
          LDI   ; copy udg3 to scr, udg2-1
          EXX
          DEC   HL                ; next udg1-1
          DEFB  #DD                ; dec ixl
          DEC   L
          JP    nline

savesp    LD    SP,0              ; retrieve stack
          LD    IY,#4000          ; repair IY
          POP   HL                ; repair EXX-reg
          POP   DE
          POP   BC
          EXX

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

x
n         EQU   101
          EQU   27

keytab    DEFB  54,38,52,53      ; QAOP

lowres    DEFB  118

```

```

score      DEFB 28,28,28,28,0
lives      DEFB 28,0
text       DEFB "G"+x,"H"+x,"O"+x,"S"+x,"T"+x,"B"+x
           DEFB "U"+x,"S"+x,"T"+x,"E"+x,"R"+x,"S"+x,0
power      DEFB 28,28,0
hiscore    DEFB 28,28,34,28
           DEFB 118

space      EQU   hrstack-$

FREE       EQU   space-sr

           DEFS  FREE

init       LDIR                      ; repair 48K bug
           LD    HL,setinit
           LD    DE,#4000             ; set screensetup and
           LD    C,36
           LDIR                      ; 2 udg over sysvar
           LD    SP,hrstack
           LD    C,10                 ; code is already executed

           LD    DE,hrstack           ; 00
           LD    HL,dispstack         ; 03
           LDIR                      ; 06
           JP    #4000                ; now go to stack area

setinit    LD    HL,hrstack           ; 00
           LD    C,100                 ; 03
           LDIR                      ; 05
           LD    (HL),1               ; 07 scf end of screen
           JP    loadstart            ; 09
           DEFB 0,0,0,0,0,0,0,0      ; 0c

           DEFB 0,192,192,224,144,208,208
udg2c      DEFB 16                     ; 23

           DEFB 0,10,12,14,9,13,13
udg1c      DEFB 1                     ; 1b

udg1       EQU   #401B                ; left man1
udg2       EQU   #4023                ; left man2

; 01      23      45      6      7      89
; flags,udg1,udg3,x1,x2,udg2
dispstack  DEFW 0,#4000,noudg         ; default setup of a line
           DEFW #0000
           DEFW #4000

vars       DEFB 128
?
last       EQU   $

```