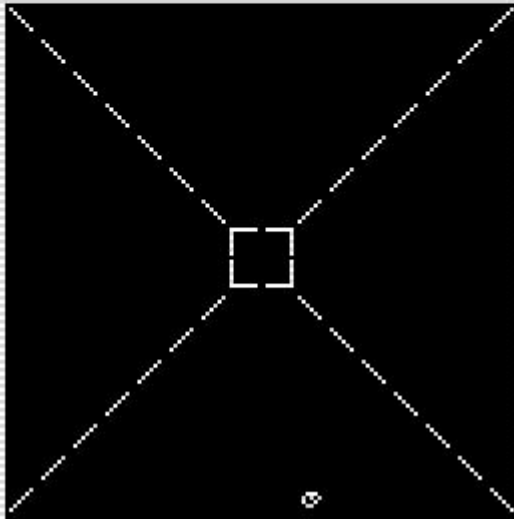


## Startrip

0000 **STARTRIP** 97 0057  
RIGHT



**I was trying some kind of display and it ended in this. The gameplay I had in mind was not so easy to code. The game was delayed due to that. In the end I got a good gameplay.... and 13 bytes left.**

```
; 3D Startrip  
; Game 57 in 1K hires for the ZX81
```

```
? * TORNADO *
```

```
sr          EQU   32                ; size of stack during game  
  
            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  0  
stkbot      DEFW  last               ; needed to load  
stkend      DEFW  last               ; needed to load  
berg        DEFB  0  
mem         DEFW  0  
            DEFB  0
```

```

initl      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      initl
cdflag     DEFB 64             ; used by zx81

; 6 bytes only used once, but screen on unloadable memory
; so a lot of memory saved for coding
; later this space is used to save UDG
copbuf     DEFB 0
xset0      LD      B,3         ; 02 LDI will corrupt B once
           LD      DE,init+2+70 ; 05 start of mid screen
           DEFB 195           ; 06 hidden JP #4000

udg1       DEFB 0,64,32,16,8,4,2,0

; the 4 central udg
udg3       DEFB 254,128,128,128,128,128,128,0
udg4       DEFB 128,128,128,128,128,128,254,0
udg5       DEFB 127,1,1,1,1,1,1,0
udg6       DEFB 1,1,1,1,1,1,127,0

udg7       DEFB 0,0,24,24,0      ; planets
udg8       DEFB 0,0,28,56,0,0
udg9       DEFB 0,24,46,116,24,0,0
udg10      DEFB 24,38,101,190,100,24

udg11      DEFB 0,0,8,16,0,0     ; gasclouds
udg12      DEFB 0,20,8,20,0,0
udg13      DEFB 0,20,40,20,40,0,0
udg14      DEFB 40,84,42,84,42,20

udg15      DEFB 0,0,24,16,0      ; meteorites
udg16      DEFB 0,0,0,24,36,24,0
udg17      DEFB 0,28,42,36,24,0,0
udg18      DEFB 124,130,170,65,138,124,0

udgdtab    DEFB udg15*256/256    ; meteroids
           DEFB udg16*256/256
           DEFB udg17*256/256
           DEFB udg18*256/256
u1         DEFB 0

           DEFB udg7*256/256      ; planets
           DEFB udg8*256/256
           DEFB udg9*256/256
           DEFB udg10*256/256

```

```

u2          DEFB 0

          DEFB udg11*256/256      ; gasclouds
          DEFB udg12*256/256
          DEFB udg13*256/256
          DEFB udg14*256/256
u3          DEFB 0

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

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

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

          LD     A,99           ; maxenergy
          CALL  setengy

nextitem    LD     B,4          ; gas, planet or 2x asteroid
          CALL  rnd
          JR     Z,asteroid     ; 0,1,2,3
          DEC     A             ; 0,1,2
asteroid    LD     H,udgdtab/256 ; 0,0,1,2
          LD     L,A
          ADD    A,A             ; a=2*a
          ADD    A,A             ; a=4*a
          ADD    A,L             ; a=5*a
          ADD    A,udgdtab*256/256
          LD     L,A            ; HL points to right graphics

          PUSH   HL

          LD     B,16
          CALL  rnd
          LD     C,A            ; start X of space item

          CALL  rnd
          LD     B,A            ; BC now xy of space item
          POP    DE

ploop       LD     A,4
ploop1      PUSH   AF

          LD     HL,msg
          LD     (HL),118       ; erase old message
          PUSH   DE
          LD     A,C
          INC     A              ; display is 1 more
          LD     (pos3+1),A

          PUSH   BC

```

```

LD     DE,msg
LD     HL,left
CP     8                                ; 7+1
JR     C,setmsg
LD     A,8
CP     C
LD     L,right*256/256
JR     C,setmsg
LD     L,down*256/256
CP     B
JR     C,setmsg
DEC    A
DEC    A
CP     B
JR     C,nomsg
LD     L,up*256/256
setmsg LD     BC,5
LDIR
nomsg  POP    BC
CALL   field
POP    DE                                ; save pointer
LD     A,(DE)
LD     (HL),A
LD     (orudg+1),A

PUSH   BC
PUSH   DE
PUSH   HL

CALL   sqtest
JR     NZ,nosq

LD     HL,taddr
LD     A,C
SUB    7
LD     C,A

LD     A,B
SUB    7
ADD    A,A
JR     NZ,cok                            ; bottom goes ok
XOR    1
cok    XOR    C

ADD    A,L
LD     L,A
LD     L,(HL)
INC    L
LD     A,L
LD     (orgudg+1),A
LD     DE,copbuf
LD     BC,7
LDIR                                ; save original udg
LD     L,A

orudg  LD     DE,#4000

; copy udg over cross
LD     B,7
setudg LD     A,(DE)
OR     (HL)
LD     (HL),A
INC    HL
INC    DE

```

```

        DJNZ setudg
        XOR    A

nosq    PUSH   AF
        CALL  delay
        POP   AF
        POP   HL
        LD    (HL),#12          ; erase current display

        JR    NZ,zxexpand      ; not on cross

orgudg  LD     DE,#4000         ; repair original udg
        LD    HL,copbuf
        LD    C,7
        LDIR

; player movement
zxexpand LD    BC,%1110000000000111
        LD    A,#A0
        OUT   (C),A
        LD    HL,keytab-1      ; set HL and 10 tstates delay
        IN    A,(C)
        LD    B,5

zxp2key INC    HL              ; change joystick in keypress
        ADD   A,A
        JR    NC,dirfnd        ; a joystick is used
        DJNZ  zxp2key

keyb    LD     BC,(lastk)      ; here no joystick used
        LD    A,C
        INC   A
        CALL  NZ,#7BD         ; a key is pressed
        LD    A,(HL)          ; get keycode of pressed key
        POP   DE
        POP   BC
        PUSH  BC
        LD    HL,keytab       ; the directiontable
        CP    (HL)
        INC   HL
        JR    NZ,tdown
        INC   B                ; up-"key" pressed

tdown   CP     (HL)
        INC   HL
        JR    NZ,tleft
        DEC   B                ; down-"key" pressed

tleft   CP     (HL)
        INC   HL
        JR    NZ,tright
        INC   C                ; left-"key" pressed

tright  CP     (HL)
        JR    NZ,tmove
        DEC   C                ; right-"key" pressed

tmove   LD     A,15
        CP    B
        JR    C,ermove        ; out of screen up and down
        CP    C
        JR    C,ermove        ; out of screen left and right
        POP   AF              ; drop old xy
        DEFB  254             ; hide get old xy
        POP   BC              ; get old xy

ermove  POP     AF
        DEC    A              ; inner loop same udg
        JP     NZ,ploop1

```

```

CALL enval                ; travel cost energy

INC DE
LD A,(DE)                ; test next graphic
OR A                     ; end reached?
JP NZ,ploop              ; outer loop udgs

result LD D,A              ; D now 0
LD A,E
CP u1*256/256            ; no asteroid, no energy loss
JR NZ,dzero
LD D,251                  ; preset missed asteroid
dzero CALL sqtest

JR Z,hit                  ; item in midscreen

enchange LD A,(enval+1)
ADD A,D
set0 CP 240                ; empty tank
INC A
JR NC,set0
seta DEC A
CP 100                    ; full tank
JR NC,seta
CALL setengy
nitem JP nextitem

hit LD A,E
LD D,20
CP u3*256/256
JR Z,enchange            ; gascloud adds energy
LD HL,score+3
CP u2*256/256
JR Z,plscore             ; score of planet = 1
DEFB #3A
ten LD (HL),28
DEC HL                    ; score of asteroid = 10
plscore INC (HL)
LD A,(HL)
CP 38
JR Z,ten
JR nitem

sqtest LD A,B
INC A
RRA
SUB 4
LD H,A
OR A
LD A,C
INC A
RRA
SUB 4
OR H                      ; when center now AH=0
RET

enval LD A,0
DEC A
setengy LD (enval+1),A
LD HL,energy
LD (HL),27
set10 INC (HL)
SUB 10
JR NC,set10

```

```

        ADD  A,38
        INC  HL
        LD   (HL),A           ; new enery set
        LD   A,(enval+1)
        OR   A
        RET  NZ
        POP  HL               ; drop RET
        JP   eog              ; out of energy is game over

field   LD   HL,init-6
        LD   A,B
        INC  A
        LD   DE,10
findy   ADD  HL,DE
        DEC  A
        JR   NZ,findy
        RET

delay   LD   A,245
        PUSH HL
        LD   HL,frames
        ADD  A,(HL)
wfr     CP   (HL)
        JR   NZ,wfr
        POP  HL
        RET

rnd      LD   HL,(frames)
rseed   LD   DE,0
        ADD  HL,DE
        INC  HL
        LD   A,H
        AND  #1F
        LD   H,A
        LD   (rseed+1),HL
        LD   A,(HL)
frnd    SUB  B
        JR   NC,frnd
        ADD  A,B
        RET

hr       LD   HL,lowres+#8000   ; the lowres display
        LD   BC,#339           ; 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,15              ; outline delay for hires
        DJNZ hr00

        LD   (savesp+1),SP     ; save current stack
        LD   SP,init           ; 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     HL                     ; L holds UDG2
POP     BC                     ; BC holds incoming UDG3
EXX
POP     BC                     ; positions UDG1 and UDG2
POP     HL                     ; L holds UDG1

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

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

lbuf   LD      R,A              ; display 16 columns
DEFB   #80,#80,#80,#80
DEFB   #80,#80,#80,#80
DEFB   #80,#80,#80,#80
DEFB   #80,#80,#80,#80
JP      Z,bloop                ; 48K bug
JP      NC,cloop               ; 48K bug
JP      savesp                 ; exit with extra blanc line

cloop  LD      A,(BC)           ; get data udg3
pos3   LD      E,4              ; set pos3, incoming item
LD      (DE),A                 ; set udg3 on pos3
INC     C                      ; next udg3
INC     L                      ; next udg2
EXX
INC     L                      ; next udg1
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

up      DEFB "U"-n,"P"-n,118
down    DEFB "D"-n,"O"-n,"W"-n,"N"-n,0
left     DEFB "L"-n,"E"-n,"F"-n,"T"-n,0
right    DEFB "R"-n,"I"-n,"G"-n,"H"-n,"T"-n

x       EQU    101
n       EQU    27

keytab  DEFB 54,38,52,53        ; QAOP

```



```

lowres      DEFB 118
score       DEFB 28,28,28,28,0

            DEFB "S"+x,"T"+x,"A"+x,"R"+x,"T"+x,"R"+x
            DEFB "I"+x,"P"+x,0
energy      DEFB 28,28,0

hiscore     DEFB 28,28,33,34
            DEFB 118
            DEFB 0,0,0,0,0,0,0
msg         DEFB 118,0,0,0,0,118

space       EQU   #4358-$

FREE        EQU   space-sr

            DEFS FREE

; some code on stackarea, later used by intrupt
setstack    LDIR                                ; 2 copy all lines on stack
            LD   B,A                            ; 3
            LD   DE,#011                        ; 6
            LD   HL,init+6                      ; 9
setx         DEC   E                            ; 10
            INC   D                            ; 11 now set pointers ok
            LD   (HL),D                        ; 12
            INC   HL                          ; 13
            LD   (HL),E                        ; 14
            LD   A,L                            ; 15
            ADD   A,9                          ; 17
            LD   L,A                            ; 18 next line
            DJNZ  setx                        ; 20 set all lines

            LD   HL,taddr                      ; 23 set HL for final setup
            JP   xset0                        ; 26 now only set mid part

            DEFS sr-26                        ; stack up to min size

; To save as much memory possible data will be copied
; 1) over sysvar and over displaydata
; 2) a part of init is set over not yet used stackspace
; the initialization is over before a displayintrupt occurs
; when that happens initializationcode is overwritten.
; So the routine must be ready before that happens
; Since the screen is a fixed displaystack, the stack is
; fully built. This makes unloadable memory available for
; the game and keeps a lot of memory free for coding.

init        LDIR                                ; repair 48K bug
            LD   HL,setinit
            LD   DE,#4000                      ; set screensetup and
            LD   C,34
            LDIR                                ; 2 udg over sysvar

            LD   SP,init                      ; now move SP, we fill

            LD   HL,xitreg                    ; move endline to right pos
            LD   DE,#43F8                    ; which is unloadable memory
            LD   C,8
            LDIR

            LD   HL,crossdata                ; save crossdata on sysvar
            LD   DE,taddr
            LD   C,4

```

```

LDIR

LD    HL,dispstack    ; firstline to right place
LD    DE,init         ; which is here above where
LD    C,10            ; code is already executed
LDIR

LD    HL,init         ; preset data for other
LD    C,150           ; 15 lines to built up
LD    A,16            ; preset 16 lines

JP    setstack        ; now go to stack area

crossdata  DEFB udg5*256/256
            DEFB udg3*256/256
            DEFB udg4*256/256
            DEFB udg6*256/256

setinit    LDI                ; 00
            INC E             ; 02
            INC E             ; 03
            INC E             ; 03
            INC E             ; 05
            INC E             ; 06
            LDI                ; 07
            INC E             ; 09
            INC E             ; 0a
            INC E             ; 0b
            DJNZ setinit      ; 0c
            JP    start       ; 0d

xitreg     DEFW 1,0,0,0      ; 10
            DEFB 0,2,4,8,16,32,64

spudg      EQU #4012        ; space udg
udg2       EQU #4019

dispstack  DEFW 0,udg2,spudg ; default setup of a line
            DEFW 0
            DEFW udg1

vars       DEFB 128
?
last       EQU $

```