

## 2D Cube

e f g h  
a k n F b i  
b e m c d j  
c a i o P k  
d j e l h l  
m n o P

Another ONELINER makes it to 1K hires. The display is based on THE EDGE, but now 6 screens are made during the display. To define the correct key pressed the keyroutine from POLICE PATROL is used.

```
; 2D Cube  
; Set all fields in correct order
```

```
? * TORNADO *
```

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

```
st      EQU    scrstart*256/256+1  
  
sb      EQU    udgsp*256/256  
ab      EQU    udga*256/256  
bb      EQU    udgb*256/256  
cb      EQU    udgc*256/256  
db      EQU    udgd*256/256  
eb      EQU    udge*256/256  
fb      EQU    udgf*256/256  
gb      EQU    udgg*256/256  
hb      EQU    udgh*256/256  
ib      EQU    udgi*256/256  
jb      EQU    udgj*256/256  
kb      EQU    udgk*256/256  
lb      EQU    udgl*256/256  
mb      EQU    udgm*256/256  
nb      EQU    udgn*256/256  
ob      EQU    udgo*256/256  
pb      EQU    udgp*256/256
```

```

basic      LD    B,5                ; preset for 48K bug
           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                ; CP n ; skip flagx
flagx      DEFB 0

           EX    AF,AF'            ; intruptcounter reset
           DEFB 17                ; LD DE,nn ; skip taddr

taddr      DEFW 3213                ; used by ZX
setrnd     JR    init1

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

hr         LD    B,8                ; sync hires display
hr00       DJNZ  hr00

           EXX

h1         EQU    high1*256/256

           LD    BC,4*256+h1
           LD    HL,scrtoken

           LD    IX,high+#8000
           CALL docopy              ; display top controls
           NOP
           DEFB #DD                ; point to next displaybuffer
           LD    L,C
scrloop    CALL docopy              ; show 4 game lines + controls
           DJNZ scrloop

           LD    IX,high+#8000
           EXX                    ; needed for sync display
           LD    B,#0A              ; 1 delay less to sync
           NOP
           CALL delb                ; show bottom controls

```

```

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

docopy    EXX
LD B,#0B
delb      DJNZ delb
LD DE,screen          ; start of the screen
EXX
built     LD A,(HL)      ; get UDG
INC HL          ; point to next
EXX
LD BC,#8FF        ; 8 lines to copy, C for LDI
LD H,udga/256     ; mainbyte of UDG
LD L,A            ; pointer to UDG
b1        LDI           ; copy to screen
LDI          ; 2 bytes
LD A,E            ; next line for
ADD A,15          ; next bytes of UDG
LD E,A
DJNZ b1           ; copy full UDG
XST       SUB 15*9-2   ; point to next UDG on screen
LD E,A
EXX
CP 110            ; end reached?
JR C,built        ; if not, do 6 UDG in total

;now it is built show it directly
display   EXX
LD B,17           ; 16 lines, 1 more needed
LD HL,screen      ; the start to display
d1        LD DE,17    ; pointer to next line
DEC B
RET Z            ; never true. timing only
d2        LD A,(HL)   ; timing
NOP
EX (SP),HL
EX (SP),HL
LD A,H           ; get high displaybyte
LD I,A
LD A,L           ; get low displaybyte
JP (IX)          ; go to displaybuffer

high      LD R,A
DEFB 64,64       ; skip corner
DEFB 64
DEFW 0
DEFB 64
DEFW 0
DEFB 64
DEFW 0
DEFB 64
DEFW 0
LD A,(HL)
RET Z            ; skip corner
JP low          ; 48K bug

high1     LD R,A
DEFW 0
DEFB 64
DEFW #8080       ; invert fielddisplay
DEFB 64
DEFW #8080

```

```

        DEFB 64
        DEFW #8080
        DEFB 64
        DEFW #8080
        DEFB 64
        DEFW 0
        JP   low                ; 48K bug

low      LD    A,(HL)            ; timing
        BIT   0,B
        JR    Z,d1              ; double display of line
        ADD   HL,DE              ; point to next line
        DJNZ  d2                ; do the screen

        EXX
        RET                    ; display done

solution DEFB ab,bb,cb,db,ib
        DEFB bb,eb,fb,gb,hb,jb
        DEFB cb,ib,jb,kb,lb,kb
        DEFB db,mb,nb,ob,pb

space    EQU   #4100-2-$
        DEFS  space

; the UDG's, a space first

udgsp    DEFB 0,0,0,0
        DEFW 0,0,0,0,0,0,0,0
udga      DEFB 0,0
        DEFB 15,248
        DEFB 0,12
        DEFB 31,252
        DEFB 48,12
        DEFB 48,60
        DEFB 31,236
udgb      DEFB 0,0
        DEFB 48,0
        DEFB 48,0
        DEFB 63,248
        DEFB 48,12
        DEFB 48,12
        DEFB 63,248
udgc      DEFB 0,0
        DEFB 0,0
        DEFB 31,248
        DEFB 48,12
        DEFB 48,0
        DEFB 48,12
        DEFB 31,248
udgd      DEFB 0,0
        DEFB 0,12
        DEFB 0,12
        DEFB 31,252
        DEFB 48,12
        DEFB 48,12
        DEFB 31,252
udge      DEFB 0,0
        DEFB 31,240
        DEFB 48,12
        DEFB 48,12
        DEFB 63,240
        DEFB 48,0
        DEFB 31,248

```

udgf	DEFB 0,0
	DEFB 31,240
	DEFB 48,0
	DEFB 48,0
	DEFB 63,240
	DEFB 48,0
	DEFB 48,0
udgg	DEFB 0,0
	DEFB 31,248
	DEFB 48,12
	DEFB 48,12
	DEFB 31,252
	DEFB 0,12
	DEFB 15,248
udgh	DEFB 0,0
	DEFB 48,0
	DEFB 48,0
	DEFB 63,248
	DEFB 48,12
	DEFB 48,12
	DEFB 48,12
udgi	DEFB 0,0
	DEFB 1,128
	DEFB 0,0
	DEFB 1,128
	DEFB 1,128
	DEFB 1,128
	DEFB 1,128
udgj	DEFB 0,0
	DEFB 1,128
	DEFB 0,0
	DEFB 1,128
	DEFB 1,128
	DEFB 25,128
	DEFB 15,0
udgk	DEFB 0,0
	DEFB 12,0
	DEFB 12,192
	DEFB 13,192
	DEFB 15,0
	DEFB 13,192
	DEFB 12,192
udgl	DEFB 0,0
	DEFB 12,0
	DEFB 12,0
	DEFB 12,0
	DEFB 12,0
	DEFB 15,128
	DEFB 7,128
udgm	DEFB 0,0
	DEFB 0,0
	DEFB 62,248
	DEFB 51,156
	DEFB 49,140
	DEFB 49,140
	DEFB 49,140
udgn	DEFB 0,0
	DEFB 0,0
	DEFB 55,248
	DEFB 60,12
	DEFB 48,12
	DEFB 48,12
	DEFB 48,12
udgo	DEFB 0,0

```

        DEFB 0,0
        DEFB 31,240
        DEFB 48,24
        DEFB 48,24
        DEFB 48,24
        DEFB 31,240
udgp    DEFB 0,0
        DEFB 55,248
        DEFB 56,12
        DEFB 56,12
        DEFB 55,248
        DEFB 48,0
        DEFB 48,0
        DEFB 0,0

start   LD    A,%01111111      ; game over, wait for
        IN    A,(254)          ; space
        RRA
        JR    C,start

        LD    DE,scrstart+1
        LD    HL,solution
        LD    BC,22
        LDIR                                ; copy over text

        LD    A,33              ; random move counter
        LD    (setrnd),A

gameloop LD    HL,setrnd
        DEC   (HL)
        JR    Z,playkey        ; now solve it

        LD    HL,frames        ; some delay
        LD    A,(HL)           ; to see the moves
        SUB   10
wfr      CP    (HL)
        JR    NZ,wfr

rndseed  LD    DE,0
        LD    HL,(frames)
        ADD   HL,DE
        DEC   HL
        LD    A,H
        AND   #0F
        LD    H,A
        LD    (rndseed+1),HL
        LD    A,(HL)
        AND   15                ; Key 0-15
        JR    validkey

playkey  INC    (HL)            ; undo zero step
waitup   LD    A,(lastk)
        INC   A
        JR    NZ,waitup        ; wait for release

waitdown LD    BC,(lastk)
        LD    A,C
        INC   A
        JR    Z,waitdown      ; wait for key down
        CALL #7BD             ; translate key
        LD    HL,#7D           ; from keypress
        ADD   A,L              ; to ZX81 ascii
        LD    L,A
        LD    A,(HL)
        SUB   38                ; to 0-15

```

```

        CP    #10
        JR    NC,waitup          ; false key pressed

; handle all movement
validkey LD    DE,#101
        BIT   3,A
        JR    Z,stepfnd
stepfnd LD    D,3
        LD    C,6
        BIT   2,A
        JR    Z,dfound
dfound  LD    E,C
        LD    C,1
        AND   3
        INC   A
        LD    B,A
        LD    A,st
        SUB   C
mule    ADD   A,C                ; a=a*e+st
        DJNZ  mule
        LD    C,E
        LD    E,A
stfound PUSH  DE                ; save counter
        LD    B,3                ; 3x swap in loop
        LD    D,scrstart/256
        LD    A,(DE)
        PUSH  AF                ; save 1st swap
        LD    A,E
rotate  ADD   A,C
        LD    L,A                ; point to next
        LD    H,D
        LDI                   ; (hl) > (de)
        INC   BC                ; undo DEC BC from LDI
        LD    E,A                ; point to next
        DJNZ  rotate
        POP   AF                ; get first
        LD    (DE),A            ; 1st on final position
        POP   DE
        DEC   D                ; right = 3x left
        JR    NZ,stfound

        LD    HL,solution
        LD    DE,scrstart+1
        LD    B,21
check   LD    A,(DE)
        CP    (HL)
        JP    NZ,gameloop        ; test solution found
        INC   DE
        INC   HL
        DJNZ  check

        LD    DE,scrstart+1
        LD    HL,wintext
        LD    C,22
        LDIR                   ; set win text

        JP    start              ; restart game

scrtoken EQU   $-1
DEFB eb,fb,gb,hb,0
scrstart DEFB ab,ab,bb,cb,db,ib
         DEFB bb,eb,fb,gb,hb,jb
         DEFB cb,ib,jb,kb,lb,kb

```

```

        DEFB db,mb,nb,ob,pb,lb
        DEFB 0,mb,nb,ob,pb

wintext    DEFB sb,sb,sb,sb,ib
           DEFB bb,gb,ob,ob,db,jb
           DEFB cb,jb,ob,bb,sb,kb
           DEFB db,sb,sb,sb,sb

space2     EQU  #4300+94-$
           DEFS space2

screen     EQU  $

init        LD  IX,hr           ; Hires mode
           LD  SP,#4400
           LD  H,#3F           ; #3fxx
           LD  D,#BF           ; #bfxx
           LD  E,L
           LDIR                ; repair 48K bug
           LD  DE,#4000
           LD  HL,keys
           LD  C,16
           LDIR
           JP  start

keys        DEFB 5,39,3,7       ; abcd
           DEFB 12,8,9,34       ; efgh
           DEFB 27,33,32,31     ; ijkl
           DEFB 37,38,26,25     ; mnop

vars        DEFB 128
?
last        EQU  $

```