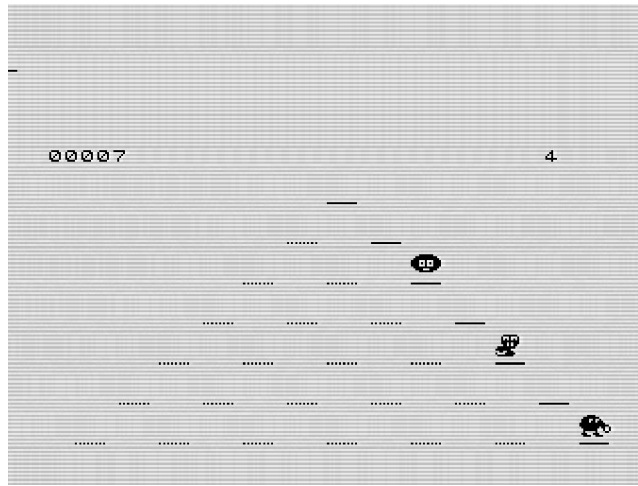


Qbert



As in the first 10 games, the final game is a charm (well I admit, the games 11 to 20 are all jewels in their own way). Qbert is a multilevel (3 levels) simplified version of this classic arcade game. Not only with 3 great graphics (Qbert, snake and ball) but also a fullscreen game to play. Only downside: The returnplatforms on the sides are not there, but we have first time run, celibration after completing a level and even a second and third level to play. After level 3 you will stay in level 3. The game was rather easy to code, although switching from graphics to field where the graphic is on cost me more time than wanted. This is the reason for the gap between a graphic and the 'cube' it stands on. The displayroutine from SALOON CITY is used. The gaps on the same row load the correct graphic to display. Due to the cube above and below you get the illusion that each position can be reached.

```
; Qbert

; multilevels

; playfield in coordinates
;           1,1
;           2,1  1,2
;           3,1  2,2  1,3
;           4,1  3,2  2,3  1,4
;           5,1  4,2  3,3  2,4  1,5
;           6,1  5,2  4,3  3,4  2,5  1,6
;           7,1  6,2  5,3  4,4  3,5  2,6  1,7

; from coordinates to field
; field array
; 1 2 2 3 3 3 4 4 4 4 5 5 5 5 5 6 6 6 6 6 6 7 7 7 7 7 7 7
; store character and field display in 1
; bit 7 used for changed field
; ld a,d 122    ; ball
; ld a,e 123    ; space
; ld a,h 124    ; Q*bert
; ld a,l 125    ; snake

; for multilevels bit 6 and 7 will be used
; ball, space, qbert and snake will be set by adding 122.

ball      EQU    0
space     EQU    3
qbert     EQU    2
snake     EQU    1
```

```

; qbert controls      check
; snake controls      check , do better
; ball controls       check
; score               check
; dead + restart      check

? * TORNADO *

                ORG   #4009                ;#4009
                DUMP  49161

                JP    begin

d_file          DEFW  dfile
dfcc            DEFW  dfile+1
var             DEFW  vars
dest            DEFW  0
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
dfsz            DEFB  2
stop            DEFW  1
lastk           DEFB  255,255,255
margin          DEFB  55

visited         EQU   #4000                ; set visitedcounter on sysvar

nxtlin          DEFW  basic
oldppc          DEFW  0
flagx           DEFB  0
strlen          DEFW  0
taddr           DEFW  3213
seed            DEFW  0
frames          DEFW  65535
snakecnt        EQU   $
coords          DEFB  0,0
prcc            DEFB  188
sposn           DEFB  33,24
cdflag          DEFB  64

lbuf            LD    A,H                ; lbuf is altered on display
                LD    R,A                ; each line 1 item is added
item1           DEFW  0                ; to the display.
                LD    A,H                ; the items to display are set
                LD    R,A                ; before the next line display
                DEFW  0                ; field on line 2
                LD    A,H
                LD    R,A
                DEFW  0                ; field on line 3
                LD    A,H
                LD    R,A
                DEFW  0                ; field on line 4
                LD    A,H
                LD    R,A
                DEFW  0                ; field on line 5
                LD    A,H
                LD    R,A
                DEFW  0                ; field on line 6
                LD    A,H
                LD    R,A

```

```

DEFW 0 ; field on line 7
JP (IX)

; The graphical characters of Q*bert
gr1 EQU graph1*256
gr2 EQU graph2*256/256
gr3 EQU graph3*256
gr4 EQU graph4*256/256
gr5 EQU graph5*256
gr6 EQU graph6*256/256

ballcnt DEFB 0
graph1 DEFB 015,000,063,192,121,032,123,112,255,248
graph5 DEFB 255,254 ; coloured 'cube'
DEFB 127,217,063,137,032,134,113,192,060,240 ; Q*bert

graph2 DEFB 015,224,017,016,021,080,031,240,029,112,013,096
DEFB 127,192,255,000,051,128,135,192,127,128 ; Snake

graph3 DEFB 015,224,063,248,127,252,241,030,245,094,241,030
DEFB 127,252,060,120,015,224 ; Ball

graph4 DEFB 000,000,000,000,000,000,000,000,000,000,000,000
DEFB 000,000,000,000,000,000,000,000,000,000 ; Space

graph6 DEFB 170,170 ; uncoloured 'cube'

field XOR A
LD H,C
LD L,B
DEFB 218 ; JP C to start at fld0
fld1 ADD A,H
INC H
fld0 DEC L
JR NZ,fld1
LD L,C
fld2 ADD A,L
DEC L
JR NZ,fld2
LD L,A
LD H,#40
LD A,(HL) ; fetch field value
AND 63 ; delete visited
CP qbert ; check with space or qbert
LD A,(HL) ; refetch value
RET

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

```

```

begin      LD      IX,hr

line1      EQU     dfile+33

          LD      B,32
          LD      HL,line1
          LD      (HL),118
resline    DEC     HL
          LD      (HL),0
          DJNZ   resline
startgame  LD      (snakecnt+1),A
          LD      HL,dfile
          LD      B,4
resscore   INC     HL
          LD      (HL),28
          DJNZ   resscore

          LD      L,line1*256/256-1
          LD      (HL),32

          LD      HL,#701
          LD      (ballxy+1),HL

gameplay   LD      E,0
          LD      A,28                ; here comes decrease counter
          LD      (visited),A
;          LD      D,n
deadin     LD      HL,#401D           ; cls fields
cls        LD      A,(HL)             ; set fields to space
          AND     E
          OR      space
          ADD     A,D                 ; add default start
          LD      (HL),A
          LD      B,L
          DEC     L
          JR      NZ,cls
nextlife   LD      HL,frames          ; some delay before
          LD      B,25                ; starting next life
          LD      A,(HL)
wfr        CP      (HL)
          JR      Z,wfr
          DJNZ   wfr-1
          INC     B                   ; start of qbert
          LD      C,B                 ; at pos 1,1
          LD      (snakexy+1),HL      ; snake out of screen
          LD      HL,#701
          LD      (ballxy+1),HL      ; ball out of screen

play       PUSH    BC
ballc      LD      A,0
          INC     A
          AND     1
          LD      (ballc+1),A
          JR      NZ,noball
          LD      HL,ballcnt
          LD      A,(HL)
          OR      240
          INC     A
          LD      (HL),A
          LD      BC,#101
          JR      Z,ballx
ballxy     LD      BC,#701
          CALL   field
          AND     192

```

```

OR    space
LD    (HL),A          ; erase old
LD    A,B
ADD   A,C
CP    8
JR    NC,noball       ; last row
CALL  rnd
INC   B
RRCA
JR    C,ballx
INC   C
DEC   B
ballx CALL field
AND   192
OR    ball
LD    (HL),A
LD    (ballxy+1),BC
noball LD A,(snakexy+1)
CP    8
JR    C,snakexy
LD    HL,snakecnt
LD    A,(HL)
OR    248
INC   A
LD    (HL),A
LD    A,7
CALL  rnd

LD    C,A
XOR   7
LD    B,A
INC   C
JR    setsnake
snakexy LD BC,0
snkcnt LD A,0
DEC   A
AND   7
LD    (snkcnt+1),A
JR    NZ,setsnake
CALL  field
AND   192
OR    space
LD    (HL),A          ; erase snake
; follow playermovement
POP   HL
PUSH  HL
LD    A,B
CP    H
JR    Z,calcdx
JR    NC,dysnake
ADD   A,C
CP    8
INC   B
JR    C,setsnake
DEC   B
calcdx LD A,C
CP    L
DEC   C
JR    NC,setsnake
INC   C
INC   C
DEFB  62
dysnake DEC B
setsnake CALL field
AND   192

```

```

OR    snake
LD    (HL),A
LD    (snakexy+1),BC
nosnmove POP BC

LD    E,8
LD    HL,frames
LD    A,(HL)
wfr1  CP    (HL)
JR    Z,wfr1
DEC    E
JR    NZ,wfr1-1
LD    A,(lastk)
INC    A
JR    Z,nokey
PUSH AF
CALL  field

LD    (HL),space        ; erase qbert
SET    6,(HL)            ; but set visited
POP    AF
CP    %11111110+1
CALL  Z,down
CP    %11110111+1
CALL  Z,left

CP    %11101111+1
CALL  Z,up
XOR    %01111111+1
dokey CALL  Z,right
nokey LD    (lastk),A        ; always impossible key
JP    NZ,play

leveldone LD    B,60        ; show some celebration
flash0 LD    HL,#401D        ; end of playfield
flash LD    A,128
XOR    (HL)                ; swap bit 8
LD    (HL),A                ; and so the display of
DEC    L                    ; all fields on pyramid
JR    NZ,flash              ; on full screen
LD    HL,frames
wfr3 LD    A,(HL)
wfr2 CP    (HL)
JR    Z,wfr2                ; wait a frame
DJNZ flash0                 ; repeat 60 flashes
JP    gameplay              ; next levelstart

addscore LD    HL,score
sc1 DEC    HL
INC    (HL)
LD    A,(HL)
CP    38
RET    NZ
LD    (HL),28
JR    sc1

up DEC    B
DEFB 62
down INC    B
DEFB 62
left DEC    C
DEFB 62
right INC    C
JR    Z,dead-1              ; left or right out of field
LD    A,8

```

```

SUB C
SUB B
JR C,dead-1 ; C set when out of bottom

CALL field ; we have a valid move
JR C,dead-1 ; something on field
DEC A ; space to qbert
LD (HL),A ; set qbert on field
ADD A,64 ; go to next field'colour'
RET C ; endstate reached
LD (HL),A ; set 'colour' as well
CALL addscore ; score a point
LD HL,visited
DEC (HL) ; counter visited must go 0
RET

dead POP AF ; drop return
LD HL,line1-1
DEC (HL)
LD A,(HL)
CP 28 ; check last life
LD E,192 ; 2 bits for field colour
JP NZ,deadin ; continue playing
LD HL,basic-1
LD DE,score+18 ; place of hiscore
LD BC,5

seekhi DEC C
JR Z,gameover ; same score as highscore
INC DE ; test next scorebyte
INC HL
LD A,(DE) ; highscore nr
CP (HL) ; compare with score
JR Z,seekhi ; not lower than highscore
JR NC,gameover ; lower score

sethi LDIR ; higher score, set new hi

gameover LD A,(lastk)
SUB %10111111 ; press ENTER to start
JR NZ,gameover ; a new game
JP startgame

hr LD B,14
hr00 DJNZ hr00

LD BC,#230
LD HL,dfile+#8000
LD A,#1E
LD I,A
LD A,#F5
CALL #2B5

LD A,#40
LD I,A

hr01 LD B,7
DJNZ hr01
LD HL,item1
LD (stpos+1),HL

LD B,7 ; set all fields no show
nshw LD (HL),H
INC L
LD (HL),H
LD A,L
ADD A,4

```

```

LD    L,A

DJNZ  nshw

LD    HL,stfld+1
LD    (HL),1          ; set first field to show

stpos LD    B,7
LD    HL,0             ; next item made visible
LD    (HL),B
INC   L
LD    (HL),B
LD    A,L
ADD   A,4

LD    (stpos+1),A      ; store next startvalue

; in displayloop set each line 1 field more

stfld LD    A,0
LD    C,7
ADD   A,C
SUB   B
LD    (stfld+1),A
LD    L,A
LD    DE,lbuf
LD    H,D
setchar LD    A,(HL)      ; convert fieldvalue
AND   3
ADD   A,122            ; to a displayudg
LD    (DE),A           ; LD A, 'DEHL'
INC   L
LD    A,5
ADD   A,E
LD    E,A
DEC   C
JP    NZ,setchar
LD    IX,cloop

LD    HL,gr1+gr4-#202  ; point registers
LD    DE,gr3+gr2-#202  ; to all possible characters

cloop LD    C,12
INC   H                ; make characterregisters
INC   H                ; point to the next line
INC   L                ; of the udg
INC   HL               ; for timing sometimes
INC   D                ; register or registerpair
INC   D                ; used.
INC   DE
INC   DE
DEC   C
JP    NZ,lbuf+#8000    ; show characters

LD    C,7
LD    HL,(stfld+1)
LD    H,#40
LD    DE,lbuf

setline LD    A,(HL)
RLCA
RLCA
AND   3

```



```

        ADD  A,122
        LD   (DE),A           ; now set bottom udg
        INC  L
        LD   A,5
        ADD  A,E
        LD   E,A
        DEC  C
        JR   NZ,setline

        EX   (SP),HL
        EX   (SP),HL
        OR   (HL)

        LD   DE,gr5+gr6
        LD   HL,gr5+gr6
        LD   IX,floop
        JP   lbuf+#8000       ; show bottom line

floop   LD   A,B
        RRCA
        JR   C,t12            ; 12 or 11 tstates in 2 lines
t11     NOP                  ; to sync display
t12     LD   C,7
        EX   (SP),HL
        EX   (SP),HL
        DEC  C
        JR   NZ,t12+2         ; sync display for next line
        PUSH HL              ; 11
        OR   (HL)            ; 7
        RET  C                ; 5
        RET  C                ; 5
        POP  HL              ; 10 = sum 38, so 2 EX (SP),HL will do
        DEC  B
        JP   NZ,stpos
;        DJNZ stpos           ; do next line

;hr5    EX   (SP),HL          ; screenfiller
;        DJNZ hr5

        CALL #292             ; back to normal displaymode
        CALL #220
        LD   IX,hr
        JP   #2A4

dfile   DEFB 118
basic   DEFB 32,32,32,32
score   DEFB 244,212,28
        DEFB 126,143,0,18

        DEFB 118
vars    DEFB 128
last    EQU  $

```