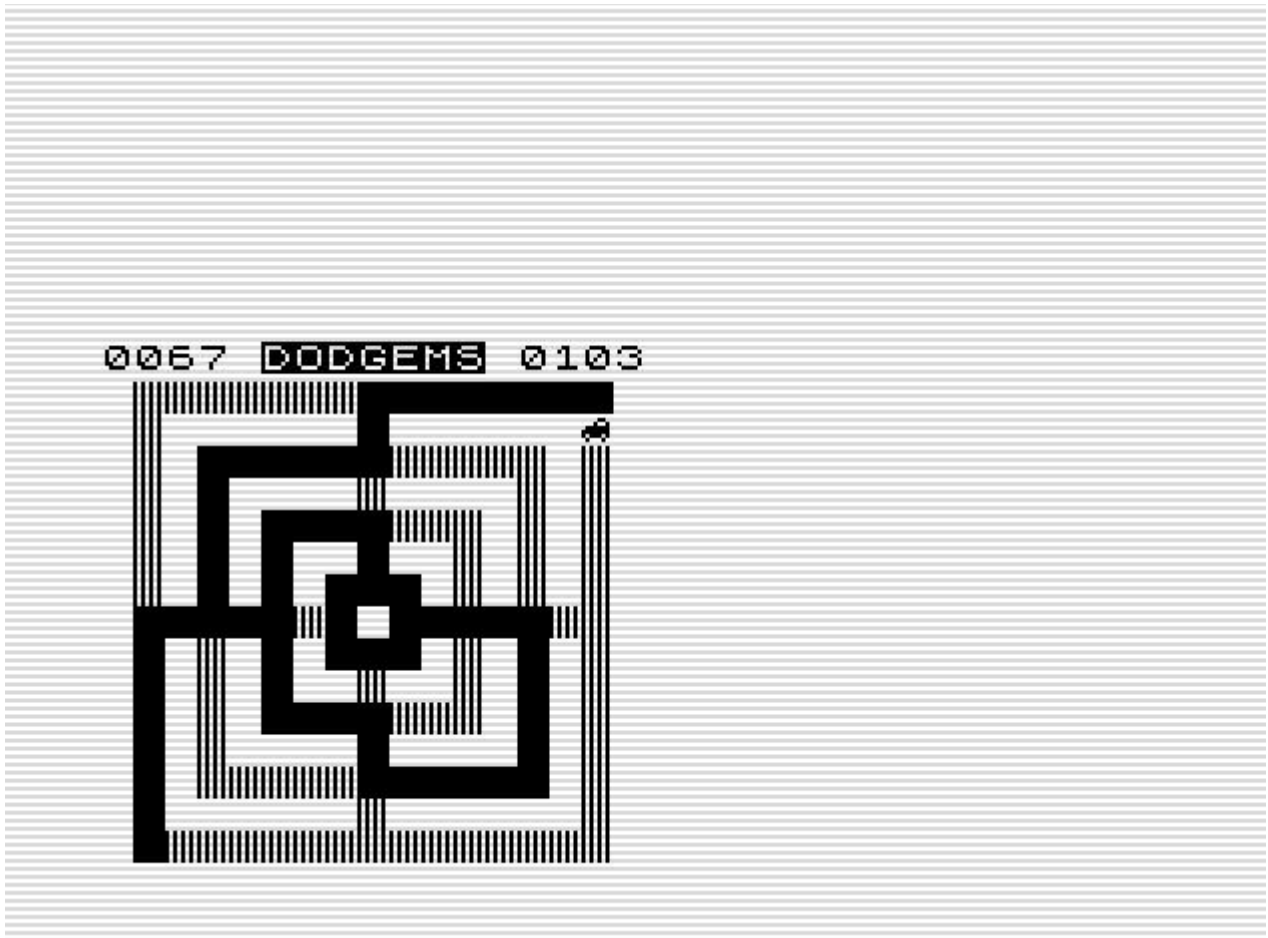


Dodgems



I tried some display to write UDG and change backgrounds. I can use this for other games too, but for now this is game 53.

```
; Dodgems
; Game 53 in 1K hires for the ZX81
; I = move in
; O = move out
; Shift is double IN or OUT
```

```
? * TORNADO *
```

```
s9      EQU  s8+16
sa       EQU  s9+16
sb       EQU  sa+16
sc       EQU  sb+16
sd       EQU  sc+16
se       EQU  sd+16
sf       EQU  se+16
```

```
u1       EQU  #7
u2       EQU  udg2*256/256
```

```
ORG  #4009          ;#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             ; 128

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

screensp DEFW #0000+u2      ; udg car comp
          DEFW #0000+u1      ; udg car player
          DEFW s1*256

          DEFW #0000+u2
          DEFW #000+u1
          DEFW s2*256

          DEFW #0000+u2
          DEFW #000+u1
          DEFW s3*256

          DEFW #0000+u2
          DEFW #000+u1
          DEFW s4*256

          DEFW #0000+u2
          DEFW #000+u1
          DEFW s5*256

          DEFW #0000+u2
          DEFW #000+u1
          DEFW s6*256

          DEFW #0000+u2
          DEFW #000+u1
          DEFW s7*256

```

```

        DEFW #0000+u2
        DEFW #000+u1
        DEFW s8*256

        DEFW #0000+u2
        DEFW #000+u1
        DEFW s9*256

        DEFW #0000+u2
        DEFW #000+u1
        DEFW sa*256

        DEFW #0000+u2
        DEFW #000+u1
        DEFW sb*256

        DEFW #0000+u2
        DEFW #000+u1
        DEFW sc*256

        DEFW #0000+u2
        DEFW #000+u1
        DEFW sd*256

        DEFW #0000+u2
        DEFW #000+u1
        DEFW se*256

        DEFW #0000+u2
        DEFW #000+u1
        DEFW sf*256

        DEFW #0000
        DEFW #0000
        DEFW 1                ; set C for return

udg2    DEFB 68,186,254,126,18,12
        DEFB 0

hr       LD    HL,lowres+#8000    ; the lowres display
        LD    BC,#241            ; minimum needed
        LD    A,#1E
        LD    I,A
        LD    A,#FB
        CALL #2B5

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

        XOR    A                ; outline delay for hires
        SCF
hr00     ADC    A,A
        JR     NC,hr00

compcarscr LD    A,(0)            ; get background computerpos
        PUSH AF                  ; save it
playcarscr LD    A,(0)
        PUSH AF
        LD     (savesp+1),SP      ; save current stack
        LD     SP,screensp       ; use display stack
        LD     A,screendata/256
        LD     I,A

```

```

        LD    D,A
        LD    H,#40
        EXX
        LD    D,A
        LD    H,#40

bloop   DEFB  #DD
        LD    L,7
        POP   BC                ; get x1 and udg1
        EXX
        POP   BC                ; get x2 and udg2
        POP   AF                ; get background pointer+flag

nline   LD    E,(HL)            ; fillers
        LD    E,(HL)
        LD    E,(HL)

        LD    E,B                ; set x2
        LD    L,C                ; set udg2 pointer
        LDI                     ; write udg to screenline
        EXX
        LD    E,B                ; set x1
        LD    L,C                ; set udg1 pointer
        LDI                     ; write udg to screenline
        JP    NC,#C008          ; do display with LBUF

savesp  LD    SP,0              ; retrieve stack
        POP   AF                ; get background computercar
        LD    HL,(playcarscr+1)
        LD    (HL),A
        POP   AF
        LD    HL,(compcarscr+1) ; get position computer car
        LD    (HL),A            ; restore background
        POP   HL
        POP   DE
        POP   BC
        EXX

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

cloop   EXX
        DEFB  #DD
        DEC   L

        NOP                     ; filler
        LD    E,(HL)            ; filler

        JR    nline

deadtest PUSH BC
        EXX
        EX    (SP),HL
        AND   A
        SBC   HL,BC            ; XY car = XY player?
        POP   HL
        EXX
        RET   NZ

        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,(lastk)       ; game over, wait for
          SUB  %10111111       ; newline
          JR   NZ,start

          LD   A,255-12         ; reset speed up
          LD   (nxtlin+1),A

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

nscreen   LD   SP,#4300        ; clear SP from dead or full
          LD   A,200           ; wait 4 sec before
          CALL nxtlin+2        ; start

loadstart LD   HL,nxtlin+1     ; speed up next level
          INC  (HL)

          LD   HL,#FF00        ; dydx for both cars
          LD   (xydir+1),HL
          LD   (xydir2+1),HL
          XOR  A
          LD   (sposn),A       ; set depth player
          LD   (sposn+1),A     ; set depth computer

          LD   HL,screensp+1
          LD   B,15

erpr      LD   (HL),A          ; erase any
          INC  HL              ; display old game
          INC  HL
          LD   (HL),A
          INC  HL
          INC  HL
          INC  HL
          INC  HL
          INC  HL
          DJNZ erpr

          LD   HL,sf+15        ; now built screen
          CP   (HL)
          JR   Z,skip
          LD   (HL),170

skip      DEC  L
          JR   NZ,makescrn

          LD   A,140           ; number of fields to
          LD   (coprcc),A      ; visit

xyplay    LD   BC,#F00        ; bottomline left
          LD   D,C
          EXX

xycomp    LD   BC,#F0E        ; bottomline right
          LD   D,C

loop      XOR  A              ; erase old display
          LD   (DE),A

```

```

LD    HL,sposn+1      ; comp depth
LD    A,(sposn)       ; player depth
SUB   (HL)
JR    Z,sethl         ; same depth,no change

SBC   A,A
ADD   A,A
INC   A               ; 255 or 1

sethl LD    H,A        ; preset in or out
LD    L,A
LD    E,A

LD    A,7
CP    C
JR    Z,cmoveb        ; on movein field vert

LD    H,0
CP    B
JR    Z,cmovec        ; on movin field hor
JR    xydir

cmoveb LD    L,0       ; dx=0
LD    A,B             ; adjust dy
CP    7
JR    C,cmove

XOR   A               ; dy=-dy
SUB   H
LD    H,A
JR    cmove

cmovec LD    A,C       ; adjust dx
CP    7
JR    C,cmove
XOR   A               ; dx=-dx
SUB   L
LD    L,A

cmove  LD    D,4
CALL  m2              ; 1 step in
CALL  m2              ; 2 step in

LD    A,(sposn+1)
ADD   A,E             ; signal in/out
LD    (sposn+1),A

xydir LD    HL,#FF00
CALL  dodydx
LD    (xydir+1),HL
JR    C,xydir

CALL  field
LD    (compcarscr+1),HL ; in HR save original back
LD    A,L
LD    (DE),A          ; set position of car

EXX
; player registers here

CALL  field
INC   DE
INC   DE
XOR   A

```

```

LD      (DE),A          ; erase old display

LD      A,%11111110    ; read shift
IN      A,(254)
RRA
LD      A,1
JR      C,double
INC     A

double   PUSH AF          ; save number of in/out
LD      HL,0
LD      A,7
CP      B
JR      Z,moveplay

CP      C
JR      NZ,dtest

CP      B
CCF
SBC     A,A
ADD     A,A
INC     A
LD      H,A
JR      input

moveplay CP      C
CCF
SBC     A,A
ADD     A,A
INC     A
LD      L,A

input    LD      A,%11011111    ; Y-P
IN      A,(254)
RRA
RRA          ; test 0
LD      E,255
JR      NC,move

RRA          ; test I
LD      E,1
JR      C,dtest

XOR     A
SUB     L
LD      L,A
XOR     A
SUB     H
LD      H,A

move     CALL m1
CALL m1

CALL deadtest

LD      A,(sposn)
ADD     A,E
LD      (sposn),A

dtest   POP AF
DEC     A

```

```

        JR    NZ,double

xydir2  LD     HL,#FF00
        LD     A,C
        ADD    A,L
        CALL   dody2
        LD     (xydir2+1),HL
        JR     C,xydir2

; player pos now on new x and y
        CALL   field
        LD     (playcarscr+1),HL
        INC    DE
        INC    DE

        LD     A,(HL)
        NOP
        CP     170
        LD     A,L
        LD     (DE),A

        CALL   Z,addscore

nosc    CALL   deadtest          ; this test is ok
        EXX
        CALL   nxtlin           ; delay loop
        JP     loop

m1      LD     D,1               ; signal playermove
m2      PUSH   BC
        LD     A,B
        ADD    A,H
        LD     B,A
        LD     A,C
        ADD    A,L
        LD     C,A
        CALL   deadtest
        PUSH   HL
        PUSH   DE
        CALL   field
        POP    DE
        JR     Z,fm1
        DEC    D
        JR     NZ,okm1          ; computermove
        LD     A,(HL)
        CP     170
        CALL   Z,addscore
okm1    POP    HL
        POP    AF
        RET

fm1     POP    HL
        POP    BC
        LD     HL,0
        LD     E,L
        RET

addscore LD     (HL),255
        LD     HL,score+4
        DEFB   #3A

ten     LD     (HL),28
        DEC    HL
        INC    (HL)

```



```

LD    A,(HL)
CP    38
JR    Z,ten
LD    HL,coprcc        ; 1 field less
DEC   (HL)
JP    Z,nscreen        ; screen filled
RET

dodydx LD    A,C        ; computer move
SUB    L                ; anticlockwise add=clockwise
dody2  PUSH BC        ; player move starts here
LD     C,A

LD     A,B
ADD    A,H
LD     B,A
CALL   deadtest

PUSH   HL
CALL   field
POP    HL

JR     NZ,ok
POP    BC

SUB    H
LD     H,L                ; dy=dx
LD     L,A                ; dx=-dy
SCF
RET

ok      POP    AF
OR     A
RET

field   LD     H,screendata/256
LD     A,B
CP     15
SBC    A,A
RET     NC                ; out of screen returns 0
LD     A,B
ADD    A,A
ADD    A,A
ADD    A,A
ADD    A,A
ADD    A,C
INC    A
LD     L,A
LD     DE,screensp+1
LD     A,B
ADD    A,A
ADD    A,B
ADD    A,A
ADD    A,E
LD     E,A
LD     A,(HL)
OR     A
RET

x       EQU    101

lowres  DEFB 118,0

```

```

score      DEFB 28,28,26,27,0

           DEFB "D"+x,"O"+x,"D"+x,"G"+x,"E"+x,"M"+x
           DEFB "S"+x,0
hiscore    DEFB 28,28,33,31
           DEFB 118

space      EQU  #4300-$
           DEFS space

screendata EQU  $

f          EQU  64

           DEFB 0

s1         LD   HL,delay
           LD   DE,nxtlin
           LDIR

           LD   HL,screendata-1
           LD   DE,sf+15
           LD   B,128
s2         INC  HL
           DEFB 0,0,0,0,0,0
           LD   A,(HL)
           DEFB 0,0,0,0,0,0
           LD   (DE),A
           NOP
s3         DEC  DE
           NOP
           DJNZ s2
           LD   HL,s2-1
           LD   (HL),B
           LD   HL,sf-1
           LD   (HL),B

           DEFB f,0,f,0
s4         DEFB f,0,f,0,0,0,0,f,0,0,0,0,f,0,f,0
s5         DEFB f,0,f,0,f,f,f,f,f,f,f,0,f,0,f,0
s6         DEFB f,0,f,0,f,0,0,f,0,0,f,0,f,0,f,0
s7         DEFB f,0,f,0,f,0,f,f,f,0,f,0,f,0,f,0
s8         JP   loadstart
           DEFB f

init       LD   HL,14000
           NOP
           LD   DE,#4000
           LD   BC,36
           LDIR
           LD   HL,#4000
           LD   DE,#C000
           LD   C,36
           LDIR
           LD   SP,#4300
           LD   C,10
           JP   s1

14000      DEFB 0 ; 4000
           DEFB 34,93,127,126,72,48
udg1       DEFB 0 ; 4007

lbuf       LD   R,A ; get displayline
           DEFB 30,45,46,30,14,13,6

```

```

DEFB 0 ; 15 positions display 4011
DEFB 24,44,28,30,29,45
DEFB 30 ; 4018
JP Z,bloop ; 48K bug
JP cloop ; 48K bug

delay LD A,255
LD HL,frames
ADD A,(HL)
wfr CP (HL)
JR NZ,wfr
RET

vars DEFB 128
?
last EQU $

```