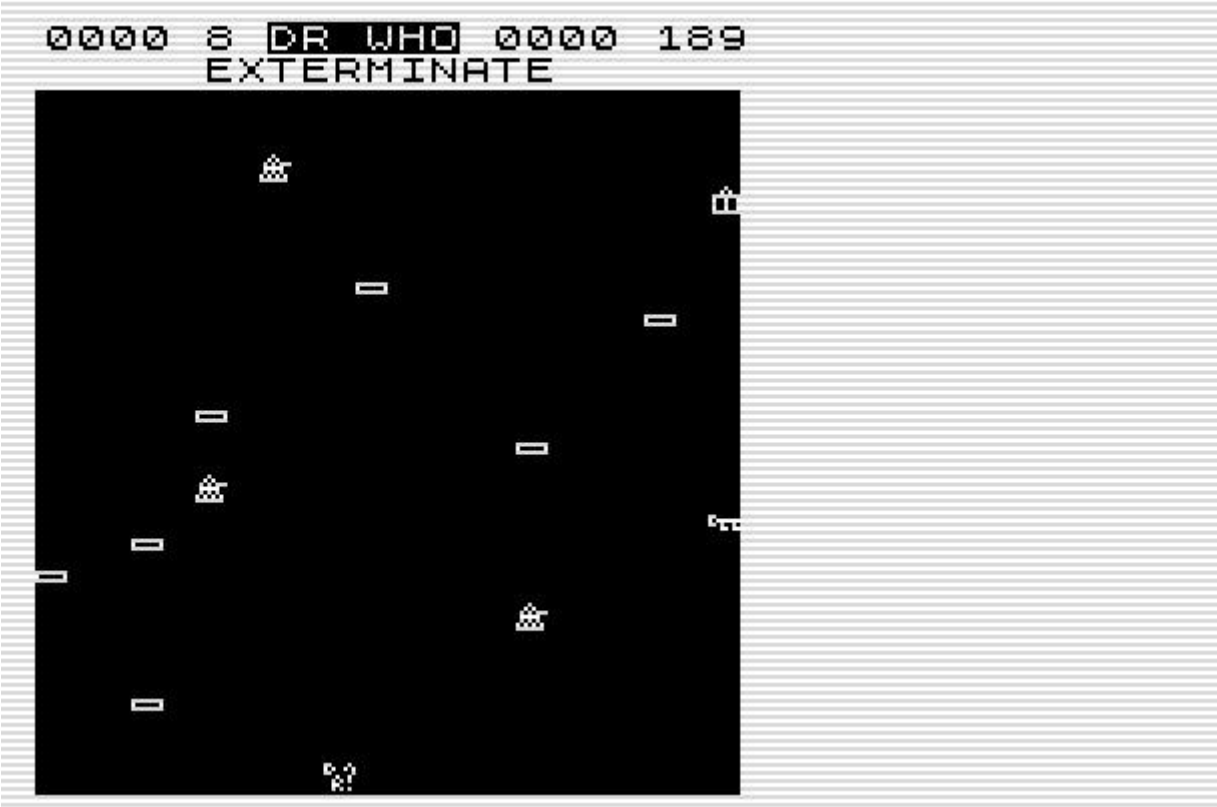
**Dr Who**




**Rod Bell asked for a Dr Who-based game. Initially it should become a puzzle, but it altered into an arcadegame. Development from previous games made the display sized 22 columns possible. Where ASTEROIDBELT had 2 udg's in 16 columns, this game has an altered method that made 22 columns possible.**

```
; Dr Who, Escape from the Daleks
; finally a 1K hires games with a Dr Who theme......

; 8 levels in 1K

? * TORNADO *

            ORG  #4009              ;#4009
            DUMP 49161

xpos        EQU  #4400-32
udgpointer EQU  initroom

basic       LD   B,5                ; preset for 48K bug
            JR   init0

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

eline       DEFW last               ; needed by loading
chadd       DEFW last-1
xptr        DEFW 0
stkbot      DEFW last
stkend      DEFW last
berg        DEFB 0
mem         DEFW 0

keys        DEFB 128
```

```
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      XOR  A                 ; delay intrupts by
           DEFB 254               ; CP n ; skip flagx
flagx      DEFB 0

           EX   AF,AF'            ; intruptcounter reset
           DEFB #3A               ; LD A,(nn) ; skip taddr

taddr      DEFW 3213              ; used by ZX81
           LD   E,L               ; low byte equal 48K bug
           DEFB #3A               ; LD A,(NN) ; skip frames

frames     DEFW 65535             ; used by ZX81
coords     JR   init1             ; useable
prcc       DEFB 188               ; used by ZX81
sposn      DEFB 33,24             ; used by ZX81
cdflag     DEFB 64                ; used by ZX81

lbuf       DEFB #80,#80,#80,#80,#80,#80,#80
           DEFB #80,#80,#80,#80,#80,#80,#80
           DEFB #80,#80,#80,#80,#80,#80,#80,#80
           JP   Z,bloop           ; 48K bug
           JP   nxtlin            ; 48K bug

daldr      DEFB 016,194           ; dalek and dr udg
           DEFB 040,165
           DEFB 087,193
           DEFB 124,18
           DEFB 084,42
           DEFB 170,48
           DEFB 254,42
           DEFB 0,0

keydr      DEFB 000,194           ; key and dr udg
           DEFB 000,165
           DEFB 192,193
           DEFB 191,18
           DEFB 210,42
           DEFB 027,48
           DEFB 000,42
           DEFB 0,0

tardr      DEFB 008,194           ; tardis and dr udg
           DEFB 020,165
           DEFB 127,193
           DEFB 73,18
           DEFB 73,42
           DEFB 73,48
           DEFB 127,42
           DEFB 0,0

pladr      DEFB 255,194           ; platform and dr udg
           DEFB 129,165
           DEFB 255,193
           DEFB 000,18
           DEFB 000,42
```

```
            DEFB 000,48
            DEFB 000,42
            DEFB 0,0

hr          LD   HL,lowres+#8000   ; the lowres display
            LD   BC,#311           ; minimum needed
            LD   A,#1E
            LD   I,A
            LD   A,#FB
            CALL #2B5

            PUSH HL                ; sync hires display
            POP  HL

            LD   (savesp+1),SP     ; save return SP
            LD   IX,nline
            LD   SP,udgpointer     ; set SP to get UDG
            LD   D,#40
            LD   A,D
            LD   I,A
            LD   BC,xpos           ; x positions of udg
            EXX
            LD   HL,lbuf+#8000
            LD   B,23              ; 22 rows

bloop       DEC  B                 ; go to next row
            RET  Z                 ; end when out of rows
            LD   C,8               ; 8 lines per row
            EXX
            POP  HL                ; get udg's to show

nline       LD   A,(BC)            ; get xpos
            LD   E,A               ; set xpos to E
            LDI                    ; copy udgdata to xpos

            LD   A,(BC)            ; get next xpos
            LD   E,A
            LDI                    ; set next xpos

            XOR  A                 ; point to start of display
            EXX
            DEC  C                 ; decrease line counter
            LD   R,A               ; set R for display
            JP   (HL)              ; start display at 1

savesp      LD   SP,0              ; retrieve SP

; fixed end of HR-routine
            CALL #292              ; back from intrupt
            CALL #220
            LD   IX,hr
            JP   #2A4

dead        LD   HL,lives          ; point to nr of lives
            DEC  (HL)              ; reduce by 1
            LD   A,(HL)
            CP   28
            JR   NZ,restlev        ; test all lives lost

            LD   HL,score-1        ; check hiscore reached
            LD   DE,hiscore-1
            LD   BC,5
fihi        INC  HL
            INC  DE
```

```
            DEC  C
            JR   Z,start
            LD   A,(DE)
            CP   (HL)
            JR   Z,fihi
sethigh     CALL C,#19F9            ; the LDIR in ROM

start       LD   A,%10111111
            IN   A,(254)
            RRA
            JR   C,start

            LD   HL,lives           ; set 8 lives
            LD   (HL),28+8

            LD   HL,score           ; reset score
            LD   B,4
ressc       LD   (HL),28
            INC  HL
            LD   A,B                ; preload A to 1
            DJNZ ressc

            LD   (loadstart+1),A    ; reset levels to 1

; screen cls can be skipped
; A short startgame clears the screen by itself
;           LD   HL,#4017
;cls        LD   (HL),b
;           DEC  L
;           JR   NZ,cls

restlev     LD   A,200              ; set energy
            LD   (cnt+1),A
            LD   SP,#4400           ; stack must be cleared HERE

loadstart   LD   B,1                ; current levelnumber
            LD   HL,level-22
            LD   DE,22
flevel      ADD  HL,DE
            DJNZ flevel
            EX   DE,HL              ; DE points to leveldata

            LD   HL,udgpointer      ; the screenudgdata
            LD   BC,xpos            ; xpos to be set
built       LD   A,(DE)             ; get leveldata
            AND  %01100000          ; make it udg only
            RRA                     ; divide by 2
            ADD  A,daldr*256/256    ; add start
            LD   (HL),A             ; set udg data
            INC  HL
            LD   (HL),#40           ; all udg in #400..
            INC  HL
            LD   A,(DE)             ; again get leveldata
            AND  %00011111          ; but now get xpos
            LD   (BC),A             ; store xpos
            DEC  BC                 ; point to next xpos
            XOR  A
            LD   (BC),A             ; on these lines no dr (yet)
            DEC  BC
            INC  DE                 ; get next data
            LD   A,(DE)             ; test if dr data is reached
            BIT  7,A                ; dr on bottom at start
            JR   Z,built            ; make all other lines
            XOR  A
```

```
            LD   (nrjump+1),A      ; reset previous jumps
            LD   (keytar+1),A      ; reset taken key
            LD   (BC),A            ; clear x item
            LD   A,(DE)
            AND  %00011111         ; make x only
            DEC  BC
            LD   (BC),A            ; set x doctor

            LD   HL,#4017          ; pointer of dalekmoves
            LD   (HL),1            ; first dalek move set
            LD   B,9               ; max 9 daleks
setdir      XOR  A
            SUB  (HL)              ; swap movedir
            INC  HL
            LD   (HL),A            ; so next dalek moves contra
            DJNZ setdir            ; do all possible daleks

playloop    LD   BC,xpos           ; the xpos of all items

            LD   DE,udgpointer     ; the kind of items
            LD   HL,#4017          ; the direction of the dalek
movedal     LD   A,(DE)            ; get kind of udg
            CP   daldr*256/256     ; test on dalek
            JR   NZ,fdal           ; if not find next
            LD   A,(BC)            ; get current xpos
            DEC  A                 ; make it 1 less
            ADD  A,(HL)            ; add movement
            CP   22                ; test below zero and above 22
            JR   C,okmove          ; in range, move ok
            XOR  A                 ; swap move direction
            SUB  (HL)
            LD   (HL),A
            JR   movedal           ; and move again

okmove      INC  A                 ; undo dec a
            LD   (BC),A            ; save new xpos of dalek
            INC  HL                ; get next direction

fdal        DEC  BC                ; get next xpos
            DEC  BC                ; get next xpos
            INC  DE                ; get next udg data
            INC  DE                ; get next udg data
            LD   A,E
            CP   hr_exit*256/256   ; test end of screen reached
            JR   C,movedal

            LD   BC,(lastk)
            LD   A,C
            INC  A
            CALL NZ,#7BD           ; translate a pressed key

            LD   D,A               ; save keycode
            LD   BC,xpos+1

fxdr        DEC  BC
            LD   A,(BC)            ; get other item on same line
            LD   L,A
            DEC  BC
            LD   A,(BC)
            OR   A
            JR   Z,fxdr            ; find position doctor

            SUB  L                 ; test on same position
            JP   Z,dead            ; horizontally dead
```

```
notdead     LD   HL,keys           ; point to keyboardcontrols

            LD   A,D              ; get keycode
            LD   E,255            ; preset for left
            CP   (HL)
            INC  HL
            JR   Z,left           ; do left
            CP   (HL)
            JR   NZ,nrjump        ; not right

            LD   E,1              ; set for right

left        LD   A,(BC)           ; get x doctor
            DEC  A                ; for test 1 less
            ADD  A,E              ; add displacement
            CP   22               ; test out of screen
            JR   NC,nrjump
            INC  A                ; undo decreasement
            LD   (BC),A           ; save new x

nrjump      LD   A,0              ; still jumping?
            OR   A

            JR   NZ,moveup        ; skip jumptest while jumping

testbot     LD   A,C
            CP   xpos*256/256-42  ; no drop at bottom
            LD   A,(BC)
            DEC  BC               ; 1 row lower, the not DR
            JR   C,jumpkey
            LD   E,A
            LD   A,(BC)
            XOR  E
            JR   NZ,droptest      ; not on something else

            CALL hittest          ; test what we have hit

jumpkey     LD   A,D              ; get keycode
            INC  HL               ; skip right test
            INC  BC
            CP   (HL)             ; test jumpkey
            JR   Z,setjump        ; jump key
            JR   noupdown         ; on something, no jump

droptest    DEC  BC
            LD   A,E              ; original x back
            LD   (BC),A           ; set dr 1 row less
            INC  BC
            INC  BC
            JR   eraseold         ; clear old position

setjump     LD   A,5              ; set 5 jumpsteps
            LD   (nrjump+1),A

moveup      LD   A,C              ; get table position
            CP   xpos*256/256-2   ; test top
            JR   NC,noupdown      ; not out of screen
            LD   A,(BC)           ; get x
            LD   E,A              ; save x
            INC  BC
            INC  BC
            INC  BC
            LD   A,(BC)           ; test x item above
```

```
            CP   E
            JR   NZ,goup           ; no block,move up possible

            CALL hittest           ; test which block
            JR   noupdown

goup        LD   A,E               ; save x
            DEC  BC
            LD   (BC),A            ; set doctor 1 line higher
            DEC  BC
            DEC  BC

eraseold    XOR  A                 ; erase old doctor position
            LD   (BC),A

noupdown    LD   HL,nrjump+1       ; decreasing stepcounter
            LD   A,(HL)
            OR   A
            JR   Z,delay           ; no decrease allowed
            DEC  (HL)

delay       LD   HL,frames         ; timing delay
            LD   A,(HL)
            SUB  8
wfr         CP   (HL)
            JR   NZ,wfr

loop        LD   A,%10001000       ; each 4 steps decrease timer
            RLCA
            LD   (loop+1),A
            JR   NC,cntmsg

cnt         LD   A,3               ; clever "short" game
            DEC  A                 ; to make correct screen
            LD   (cnt+1),A         ; when zero, loose a life
            JR   Z,dead1

            LD   HL,timer
            LD   B,100             ; set 100 counter
settime     LD   (HL),27
ft          INC  (HL)
            LD   C,A
            SUB  B
            JR   NC,ft
            INC  HL
            LD   A,B
            LD   B,10              ; next do 10 counter
            SUB  B
            LD   A,C
            JR   NZ,settime
            ADD  A,28              ; finally set 1 counter
            LD   (HL),A            ; full decreased timer set

cntmsg      LD   A,0
            DEC  A
            AND  31
            LD   (cntmsg+1),A
            JR   NZ,nomsg          ; each 32 steps swap message
            LD   HL,extmsg         ; "EXTERMINATE"
            LD   A,(HL)
            XOR  118
            LD   (HL),A

nomsg       JP   playloop
```

```
; on something, test what
hittest     LD   A,xpos*256/256+2  ; from xpos
            SUB  C                 ; to index for udgpointer
            PUSH HL
            LD   HL,udgpointer-2
fudg        INC  HL
            DEC  A
            JR   NZ,fudg
            LD   A,(HL)            ; get udgpointer
            POP  HL
            CP   pladr*256/256     ; platform is ok
            RET  Z
; check key or tardis
            CP   keydr*256/256     ; when key, set signal taken
            JR   NZ,testdal
            LD   (keytar+1),A
            XOR  A
            LD   (BC),A            ; erase key
            RET

testdal     CP   daldr*256/256     ; hit by dalek
dead1       JP   Z,dead
keytar      LD   A,0               ; tardis remains
            OR   A                 ; do I have the key?
            RET  Z                 ; no continue
nextlevel   LD   A,(cnt+1)         ; remaining time
            LD   B,A               ; add as score
addpoint    LD   HL,score+4
            DEFB 17
setnext     LD   (HL),28
            DEC  HL
            INC  (HL)
            LD   A,(HL)
            CP   38
            JR   Z,setnext
            DJNZ addpoint

nround      LD   HL,loadstart+1
            LD   A,(HL)            ; get current level
round       INC  A                 ; add 1
            LD   (HL),A
            SUB  8                 ; not more than 8 levels
            JR   Z,round
            JP   restlev           ; play next level

; 000 = dalek
; 001 = key
; 010 = tardis
; 011 = platform

; manually levels converted

level       DEFB %01100000         ; level 1
            DEFB %01100000
            DEFB %00000100
            DEFB %01010110
            DEFB %01100000

            DEFB %01100000
            DEFB %01101011
            DEFB %01110100
            DEFB %01100000
            DEFB %01100000
```

```
            DEFB %01100110
            DEFB %01110000
            DEFB %00001010
            DEFB %00110110
            DEFB %01100100

            DEFB %01100001
            DEFB %00001100
            DEFB %01100000
            DEFB %01100000
            DEFB %01100100

            DEFB %01100000
            DEFB %10001010

level2      DEFB %00110100
            DEFB %00001010
            DEFB %01000001
            DEFB %01100010
            DEFB %01110100

            DEFB %01100110
            DEFB %01110010
            DEFB %00010100
            DEFB %01101001
            DEFB %01101111

            DEFB %01100110
            DEFB %00001010
            DEFB %01110000
            DEFB %01101000
            DEFB %00010001

            DEFB %01101110
            DEFB %01101010
            DEFB %00001110
            DEFB %01101100
            DEFB %01100110

            DEFB %00010101
            DEFB %10010100          ; final is always the DOCTOR

level3      DEFB %00000110
            DEFB %00100110
            DEFB %01100000
            DEFB %01001010
            DEFB %01100000

            DEFB %00010001
            DEFB %01100000
            DEFB %01101010
            DEFB %01100000
            DEFB %00000101

            DEFB %01100000
            DEFB %01101010
            DEFB %01100000
            DEFB %00010001
            DEFB %01100000

            DEFB %01101010
            DEFB %01100000
            DEFB %00000101
```

```
            DEFB %01100000
            DEFB %01101010

            DEFB %01100000
            DEFB %10010100          ; final is always the DOCTOR

; a tooling in EXCEL gives the levelcodes now

level4      DEFB 96,33,75,11,100
            DEFB 105,110,115,11,96
            DEFB 118,11,100,105,110
            DEFB 115,96,97,11,96
            DEFB 96,139

level5      DEFB 35,84,115,102,11
            DEFB 112,99,11,115,102
            DEFB 11,112,99,11,115
            DEFB 102,11,112,99,96
            DEFB 96,139

level6      DEFB 96,75,109,96,11
            DEFB 112,96,15,115,102
            DEFB 118,50,99,2,115
            DEFB 102,2,96,99,96
            DEFB 96,139

level7      DEFB 43,13,8,111,102
            DEFB 98,115,96,96,99
            DEFB 114,96,96,102,112
            DEFB 11,96,107,4,96
            DEFB 65,129

level8      DEFB 35,3,3,3,3
            DEFB 99,83,3,113,3
            DEFB 102,96,114,96,3
            DEFB 104,113,3,3,105
            DEFB 110,139

x           EQU  101
n           EQU  27

lowres      DEFB 118
score       DEFB 0,"L"+x,"R"+x,"F"+x,0
lives       DEFB 29,0
            DEFB "D"+x,"R"+x,128,"W"+x,"H"+x,"O"+x,0
hiscore     DEFB 28,28,28,28,0
timer       DEFB 28,28,30
            DEFB 118
extmsg      DEFB 118
            DEFB 0,0,0,0
            DEFB "E"-n,"X"-n,"T"-n,"E"-n,"R"-n,"M"-n
            DEFB "I"-n,"N"-n,"A"-n,"T"-n,"E"-n
            DEFB 118

; init fully placed on leveltable
; this table is set by unpacking a level

initroom    LD   (HL),C             ; clear xpos with
            DEC  HL                 ; printable positions
            DJNZ initroom           ; to prevent programkill

            LD   HL,score+1         ; keydef table
            LD   DE,keys            ; destination defined keys
```

```
wup         LD   A,(lastk)         ; wait for no keypress
            INC  A
            JR   NZ,wup
wdown       LD   BC,(lastk)        ; get keypress
            LD   A,C
            INC  A
            JR   Z,wdown
            PUSH HL                ; save pointers
            PUSH DE
            CALL #7BD              ; translate key
            POP  DE                ; get pointers
            POP  HL
            LD   (DE),A            ; save key selected
            XOR  A
            LD   (HL),A            ; clear old key
            INC  DE                ; point to next key position
            INC  HL                ; point to next definition key
            OR   (HL)
            JR   NZ,wup            ; check end reached

            JP   loadstart         ; start ingame to built screen

            DEFW pladr             ; only NOT set item by level

hr_exit     DEFW savesp            ; end of hr routine

; BASIC initialization done on the screen
screen      EQU  $

cloop       EXX                    ; 40 this table will be
            INC  C                 ; 41 copied over sysvar
            INC  C                 ; 42 thus saving 5 bytes
            JP   (IX)              ; 44

init        LD   IX,hr             ; 04 Hires mode
            LD   SP,#4400          ; 07
            LD   H,#3F             ; 09 #3fxx
            LD   D,#BF             ; 11 #bfxx
            LDIR                   ; 13 repair 48K bug

            LD   HL,cloop          ; set a routine over sysvar
            LD   DE,nxtlin         ; 19
            LD   C,5               ; 21
            LDIR                   ; 23

            LD   B,xpos-screen     ; xpos must have a correct
            LD   HL,xpos           ; value that will work
            JR   initroom          ; before starting the game

vars        DEFB 128
?
last        EQU  $
```