

# WATERWORM

Waterworm is the classic snake (nibbles for Qbasic) readapt for MSX-BASIC v1.0 in just 10 lines for the TEN LINER CONTEST 2020.

## INSTRUCTIONS

Instructions are very easy to learn. Use arrows in the keyboard to drive the waterworm and press space to start, restart and in order to go to next level.

It is possible to play anywhere, real machines or emulators, but the most easy way of playing is to go to <https://msxpen.com/> an amazing online emulator created by Rafael Jannone (thanks from here), choose MSX Europe (PAL) in settings then select machine, copy and paste the code inside waterworm.txt into BASIC Code box and run... and enjoy!!!

## ZIP CONTENT

- waterworm.pdf - this little manual
- waterworm.dsk - game disk (files & load"waterwor.bas")
- waterworm.bas - basic code game
- waterworm.txt - code in txt format

## LEARNING CODE

```
LINE 10                                ;Configure graphics, mode Screen and static vars
DEFINT A-Z                             ;make integer all vars to get code faster
DIM M(15)                              ;the M() array has address of every chunk worm
A=256                                  ;A (& B) is a piece of address of code ASCII 32 for water fx
D=6144                                 ;name address (first byte of screen memory position)
COLOR 7,4,1                           ;set color ink, background and border by default
KEY OFF                                ;remove function keys
SCREEN 1,0,0                           ;set screen mode
WIDTH 32                               ;set screen width (32 characters)
FOR I=384 TO 728                       ;loop for improve char set
VPOKE I,VPEEK(I)ORVPEEK(I)/2          ;read VRAM address for numbers and letters and make wider
NEXT                                   ;end loop
VPOKE 8198,180                         ;set colour for some characters every 8 caracters the same colour
VPOKE 8199,180                         ;that is MSX Screen 1 limitation :(
VPOKE 8192,132                         ;
VPOKE 8196,84                          ;
VPOKE 8194,212                         ;
VPOKE 8195,25                          ;
FOR I=64 TO 111                       ;set graphics for worm from datas
READ X                                ;Read from data
VPOKE I,X                              ;set VRAM valor
NEXT                                   ;

LINE 20                                ;Title
CLS                                    ;Clear Screen
LOCATE 11,10                          ;this line doesn't need so much explanation,
PRINT "WATERWORM"                     ;basically it is for make up the program giving a initial screen
FOR I=6432 TO 6463                    ;print a couple of lines up and down of main title straightly in RAM
VPOKEI,23                             ;
VPOKEI+64,23                          ;
NEXT                                   ;
LOCATE 2,3                             ;
PRINT"BASIC 10 LINER CONTEST 2020"    ;
LOCATE6,20                             ;
PRINT"by Jos'b from Spain"           ;

LINE 30                                ;Ready to start and launch game
IFNOT(STRIG(0))GOTO30 ELSE CLS         ;wait for pressing space to start the game
S=0                                    ;S saves the value to jump the next VRAM address of worm head
C=0                                    ;C has the value of the heart to pick up
L=2                                    ;worm length (initially at 2)
F=8                                    ;worm face (8 for up, 9 for right, 10 for down & 11 for left)
P=6512                                ;memory position for worm head at beginning of level
```

```

:M(0)=P ;memory position for worm head in M() array in game
:M(1)=P ;memory position for second chunk of worm in M() array in game
:M(2)=P ;memory position for third chunk of worm in M() array in game
:VPOKE A,0 ;A & B works each other for water scroll effect
:VPOKE B,0 ;A puts water wave, and B removes water wave
:FOR I=D TO D+31 ;put wall border (outside rectangle)
:VPOKE I,28 ;
:NEXT ;
:FOR I=0 TO 23 ;
:VPOKE D+I*32,28 ;
:VPOKE D+I*32+31,28 ;
:NEXT ;
:FOR I=D+736 TO D+767 ;
:VPOKE I,28 ;
:NEXT ;
:PLAY "v11t255m4100" ;set up play intruction for emitting simple notes

LINE 40 ;Put random objects for increasing difficulty
IF V THEN FOR I=1 TO V*4 ;if level>1 then put 4 blocks by level in screen (v=current level)
:X=INT(704*RND(-TIME)+32 ;every block is chosen by random function
:VPOKE D+X,28 ;put blocks of 2x2 chars
:VPOKE D+X+1,28 ;
:VPOKE D+X+32,28 ;
:VPOKE D+X+33,28 ;
:NEXT ;

LINE 50 ;Main game loop from 50 to 80: waterworm control
VPOKE B,0 ;make water effect
:VPOKE A,49 ;
:VPOKE M(L),32 ;remove last chunk of worm
:VPOKE M(L-1),13 ;put worm tail
:VPOKE M(1),12 ;put last chunk of worm
:VPOKE P,F ;put worm head
:OS=S ;save last s (OS=old S)
:Z=STICK(0) ;read stick (from keyboard)
:S=32*(Z=1)-(Z=3)-32*(Z=5)+(Z=7)-OS*(Z=0) ;set worm direction by logical operations (see *note)
:P=P+S ;add S to current P
:E=-VPEEK(P)*(S<>0)-32*(S=0) ;save next VRAM position of worm (for later)
:F=-8*(S=-32ORS=0)-9*(S=1)-10*(S=32)-11*(S=-1) ;set head according worm direction (see **note)

LINE 60 ;Main game loop from 50 to 80: put heart if necessary
OC=C ;Old C (C holds the heart memory position)
:C=- (C=0)*(INT(RND(-TIME)*768)+6144)-OC*(C<>0) ;set heart memory position by random function
;check whether there is something or not
:J=VPEEK(C) ;if there is nothing continue game, else repeat random function
:IFJ<>32ANDOC=0THENC=0:GOTO60

LINE 70 ;Main game loop from 50 to 80: water FX and check pick heart up
VPOKE C,3 ;put heart in screen
:B=A ;make B=A
:A=A+1+8*(A=263) ;update A value rolling every 8 lines for water wave simulation
:FOR I=L TO 0 STEP-1 ;update M() array for holding VRAM worm address
:M(I+1)=M(I) ;
:NEXT ;
:M(0)=P ;M(0) holds VRAM worm head address
:J=VPEEK(P) ;check if worm eats heart
:IF J=3 THEN C=0 ;if so make C=0 to get a new heart in screen
:L=L+1 ;increase points
:PLAY"eg" ;play two simple notes
:IF L=15 THEN ;check wheter worm ate the last heart
:VPOKE M(L),32 ;if so update worm position
:VPOKE M(L-1),13 ;
:VPOKE M(1),12 ;
:VPOKE P,F ;
:V=V+1 ;increase level
:R=R+L*10 ;update points
:LOCATE 11,10 ;send message of ending level
:PRINT "LEVEL";V ;V has level
:LOCATE 11,11 ;
:PRINT"SCORE";R ;R has current points
:GOTO30 ;go to line 30 where reset values for starting next level

LINE 80 ;Main game loop from 50 to 80: check worm crashes
IF (E=32 OR E=3) ;check if worm crashes
GOTO 50 ;if not, go to main game loop in line 50 (keep on playing)
ELSE PLAY"gc" ;if so, play sadly notes of ending game
:VPOKE M(L),32 ;update worm position
:VPOKE M(L-1),13 ;
:VPOKE M(1),12 ;
:VPOKE P,F ;
:LOCATE 11,10 ;send "GAME OVER" and final points
:PRINT "GAME OVER" ;
:LOCATE 11,11 ;
:PRINT"SCORE";R+L*10 ;

LINE 90 ;Wait for restart game
IF NOT(STRIG(0)) ;wait for pressing space
GOTO 90 ;if not, keep on waiting
ELSE R=0 ;if so, set points and level zero...

```

```

:V=0
:GOTO 20
;
;...and go to title screen in line 20

LINE 100
DATA
60,126,219,153,255,189,153,129,
252,22,51,127,127,51,22,252,
129,153,189,255,153,219,126,
60,63,104,204,254,254,204,104,
63,0,24,36,66,66,36,24,
0,0,0,24,36,36,24,0,0
;Graphics datas
;datas for worm graphics
;up head (CODE ASCII 8)
;right head (CODE ASCII 9)
;down head (CODE ASCII 10)
;left head (CODE ASCII 11)
;chunk of worm (body) (CODE ASCII 12)
;worm tail (CODE ASCII 13)

;for heart the code uses the code ASCII 3
;and for walls the code ASCII 28

```

#### **\*note**

This piece of code

```
S=32*(Z=1)-(Z=3)-32*(Z=5)+(Z=7)-OS*(Z=0)
```

is similar to this one in MSX-BASIC

```

IF Z=1 THEN S=-32
IF Z=3 THEN S=1
IF Z=5 THEN S=32
IF Z=7 THEN S=-1

```

Afterwards S is added to P for moving the worm.

Take note about MSX uses &FF to TRUE representation which is -1 value, that's why we need to change the sign in the usual MSX-BASIC code. Apart from that, OS\*(z=0) is used to keep on the worm movement when not pressing any key.

#### **\*\*note**

```
F=-8*(S=-32 OR S=0)-9*(S=1)-10*(S=32)-11*(S=-1)
```

This logical operation, as **\*note** shows, can be difficult to see for people who not has programming experience enough, but it is easy to understand when translating to BASIC code

```

IF S=-32 OR S=0 THEN F=8 (up head)
IF S=1 THEN F=9 (right head)
IF S=32 THEN F=-10 (down head)
If S=-1 THEN F=11 (left head)

```

F holds the head graphics for being printed.

Badajoz 21 de febrero de 2020  
 José Javier Franco Benítez (Fran Chesstein)

Thanks for playing