

# Stroq - A game in Forth#

## Table of Contents

- [Stroq - A game in Forth](#)
- [About Stroq:](#)
- [Screenshots](#)
- [Atari 8bit Version](#)
- [License:](#)
- [Source:](#)
- [Atari XL/XE Computer \(6502 X-Forth / FIG-Forth\)](#)
- [Downloads](#)

## About Stroq:#

Stroq is a simple but addictive puzzle game. I found the game for Linux and MacOS X on the Internet ([Stroq Homepage](#)) and decided to do a conversion for the Atari 8Bit Computer in Forth. Winfried Piegsda did the graphical design, sound and artwork, I did the coding in Forth (X-Forth for Atari). The Game programmed in little less than 4 weeks to be submitted to the annual ABBUC Software Contest where it scored the [5th place](#).

## Screenshots#

## Atari 8bit Version#

## License:#

This game is licensed under the GNU Public License, Version 2 or higher.

## Source:#

## Atari XL/XE Computer (6502 X-Forth / FIG-Forth)#

I didn't look at the C-Sources, we only "stole" the puzzle level from the [original game at sourceforge](#). Because the game was programmed under time pressure, the code is not "optimal".

```
( Stroq )
```

```
CR ." Loading Stroq" CR
```

```
HEX
```

```
: AT-X ( xp -- ) 55 ! ;  
: AT-Y ( yp -- ) 54 C! ;  
: AT ( xp yp -- ) AT-Y AT-X ;  
: AT-X? ( -- xp ) 55 @ ;
```

```
58 CONSTANT SAVMSC ( Pointer to Displaybuffer )
```

```

14 CONSTANT MAXX
14 CONSTANT MAXY

0 VARIABLE pf MAXX MAXY * ALLOT
pf CELL+ pf !
pf VARIABLE pfp

." Playfield is at $" pfp @ U. CR

0 VARIABLE curx
0 VARIABLE cury

0 VARIABLE lastx
0 VARIABLE lasty
0 VARIABLE lastblox

0 VARIABLE STROQPTR
7000 CONSTANT STROQMEM

0 VARIABLE LevelWidth
0 VARIABLE LevelHeight

0 VARIABLE BloxUsed
0 VARIABLE PuzzleNumber

0 VARIABLE Xoffset
0 VARIABLE Yoffset

: WAITKEY KEY DROP ;

: [PF"]
  R COUNT DUP 1+ R> + >R
  pfp @ SWAP CMOVE
  LevelWidth @ pfp +!
;

: BEGINLEVEL
  pf @ pfp !
  0 STROQPTR !
  0 lastx ! 0 curx !
  0 lasty ! 0 cury !
  0 lastblox ! ;

: PF"
  22 STATE @ IF COMPILE [PF"] WORD
  C@ 1+ ALLOT ELSE WORD DUP C@ 1+
  PAD SWAP CMOVE PAD COUNT THEN ; IMMEDIATE

: CLS 7D EMIT ;

: CONVERTLEVEL
  pfp @ DUP
  LevelWidth @ LevelHeight @ * + SWAP DO
  I C@
  DUP 58 = IF 41 I C! THEN
  DUP 20 = IF 01 I C! THEN
  4F = IF 81 I C! THEN
  LOOP ;

```

```

: ENLEVEL
  10 LevelHeight @ - 2 / 28 * YOffset !
  28 LevelWidth @ 2 * - 2 / XOffset !
  LevelWidth @ 2 MOD 0=
  IF -1 XOffset +! THEN
  LevelHeight @ 2 MOD 0=
  IF -28 YOffset +! THEN
  pf @ pfp ! CONVERTLEVEL ;

: PRINTBLOX ( x y n -- )
  ROT 2 * SAVMSC @ + XOffset @ +
  ROT 50 * YOffset @ + + 28 -
  SWAP 2 * 40 +
  OVER OVER SWAP C!
  1+ SWAP 1+ OVER OVER C!
  27 + SWAP 3F + SWAP OVER OVER C!
  1+ SWAP 1+ SWAP C! ;

: PRINTBLOX2 ( x y )
  OVER OVER LevelWidth @ * + pfp @ +
  C@ DUP 80 AND IF 0F AND 70 + PRINTBLOX ELSE
  DUP 40 AND IF 0F AND PRINTBLOX ELSE
  DUP F0 AND 0= IF 0F AND 30 + PRINTBLOX ELSE
  DROP THEN THEN THEN ;

: PRINTLEVEL
  LevelHeight @ 0 DO
  LevelWidth @ 0 DO
  I J PRINTBLOX2
  LOOP
  LOOP ;

: AND! ( n addr -- )
  DUP C@ F0 AND ROT OR SWAP C! ;

: PUTBLOX ( x y n -- )
  ROT pfp @ + ROT LevelWidth @ * +
  AND! ;

: INLINE? ( -- f )
  curx @ lastx @ - ABS 1 =
  cury @ lasty @ - ABS 1 = + 1 = ;

: CLEARHEADER BC40 28 ERASE ;

: FLIPBLOX ( x y )
  LevelWidth @ * + pfp @ + DUP
  C@ DUP 40 AND 0= IF
  80 XOR SWAP C!
  ELSE
  DROP DROP
  THEN ;

: GETBLOX ( x y -- n )
  LevelWidth @ * + pfp @ + C@ 0F AND ;

: GETCOLOR ( x y -- )
  LevelWidth @ * + pfp @ + C@ 80 AND ;

```

```

: EMPTY? ( -- f )
  curx @ cury @ GETBLOX 1 = ;

: PUTLBLOX ( n -- )
  lastx @ lasty @ ROT PUTBLOX ;

: PUTCBLOX ( n -- )
  curx @ cury @ ROT PUTBLOX ;

: SUCCESS ( -- )
  lastx @ lasty @ PRINTBLOX2
  curx @ cury @ PRINTBLOX2
  curx @ lastx !
  cury @ lasty !
  2 STROQPTR +! 1 BloxUsed +!
  curx @ STROQMEM STROQPTR @ + C!
  cury @ STROQMEM STROQPTR @ + 1+ C! ;

: STROQADDR STROQMEM STROQPTR @ + ;
: STROQX STROQADDR C@ ;
: STROQY STROQADDR 1+ C@ ;
: REWIND
BEGIN
  STROQX STROQY 1 PUTBLOX
  STROQX STROQY PRINTBLOX2
  STROQX curx @ =
  STROQY cury @ =
  AND 0=
WHILE
  -2 STROQPTR +!
  -1 BloxUsed +!
REPEAT
-2 STROQPTR +! -1 BloxUsed +!
STROQX lastx !
STROQY lasty !
STROQPTR @ 0 > IF
  STROQX curx @ = IF
  STROQY cury @ < IF
    3 PUTCBLOX 3 lastbiox !
  ELSE
    5 PUTCBLOX 5 lastbiox !
  THEN
THEN
STROQY cury @ = IF
STROQX curx @ > IF
  4 PUTCBLOX 4 lastbiox !
ELSE
  6 PUTCBLOX 6 lastbiox !
THEN
THEN
ELSE
  2 PUTCBLOX 2 lastbiox !
THEN SUCCESS
-2 STROQPTR +!
;

: SOLVE
CLEARHEADER A 0 AT

```

```

STROQPTR @ IF
  STROQPTR @ 0 DO
    STROQX STROQY FLIPBLOX
    STROQX STROQY PRINTBLOX2
    -2 STROQPTR +!
  2 +LOOP
  0 LevelHeight @ 1 - 1 DO
    0 LevelWidth @ 1 - 1 DO
      I J GETCOLOR
      IF 2 OR ELSE 1 OR THEN
        LOOP
        3 = IF 1 OR THEN
        LOOP
        IF
          ." Puzzle Not Solved! " 0
        ELSE
          ." Puzzle Solved! " 1
        THEN
      ELSE
        ." No Stroke! " 0
      THEN WAITKEY
;

: SETSTROQ ( -- )
  INLINE? EMPTY? AND IF
    curx @ lastx @ = IF
      cury @ lasty @ - 1 = IF
        ( one down )
        lastblox @
        DUP 2 = IF 5 PUTLBLOX THEN
        DUP 3 = IF 8 PUTLBLOX THEN
        DUP 4 = IF C PUTLBLOX THEN
          6 = IF A PUTLBLOX THEN
        3 PUTCBLOX 3 lastblox !
      ELSE
        ( one up )
        lastblox @
        DUP 2 = IF 3 PUTLBLOX THEN
        DUP 4 = IF B PUTLBLOX THEN
        DUP 5 = IF 8 PUTLBLOX THEN
          6 = IF 9 PUTLBLOX THEN
        5 PUTCBLOX 5 lastblox !
      THEN
    ELSE
      curx @ lastx @ - 1 = IF
        ( one right )
        lastblox @
        DUP 2 = IF 4 PUTLBLOX THEN
        DUP 3 = IF B PUTLBLOX THEN
        DUP 5 = IF C PUTLBLOX THEN
          6 = IF 7 PUTLBLOX THEN
        6 PUTCBLOX 6 lastblox !
      ELSE
        ( one left )
        lastblox @
        DUP 2 = IF 6 PUTLBLOX THEN
        DUP 3 = IF 9 PUTLBLOX THEN
        DUP 4 = IF 7 PUTLBLOX THEN
          5 = IF A PUTLBLOX THEN

```

```

        4 PUTCBLOX 4 lastblox !
    THEN
    THEN SUCCESS
ELSE
    EMPTY? 0= IF
        REWIND SUCCESS
    ELSE
        CLEARHEADER
        E 0 AT ." Illegal Move!"
        WAITKEY CLEARHEADER
    THEN
THEN
;

: SETBLOX ( -- )
    lastblox @
    IF
        SETSTROQ
    ELSE
        curx @ cury @ 2 PUTBLOX
        2 lastblox ! SUCCESS
    THEN ;

: CUROFF FF 2F0 C! ;
: CURON 00 2F0 C! ;

: HEADER ( Print header )
    DECIMAL CUROFF
    1 0 AT BloxUsed @ S>D
    <# # # # #> TYPE SPACE ." Blox used"
    16 0 AT ." Puzzle Number "
    PuzzleNumber @ S>D
    <# # # # #> TYPE
    HEX ;

00 VARIABLE FOOTERMEM 28 ALLOT
." Footermem is at " FOOTERMEM . CR

( Displaylistgeraffel )

230 CONSTANT SDLSTL
200 CONSTANT VSDLST
D40E CONSTANT NMIEN
00 VARIABLE DLSAVE
D40B CONSTANT VCOUNT

: SAVEDL SDLSTL @ DLSAVE ! ;
: RESTORED DLSAVE @ SDLSTL ! ;
: SETDL ( addr -- )
    BEGIN
        VCOUNT C@ 10 <
    UNTIL
        SDLSTL ! C0 NMIEN C! ;

0 VARIABLE DLGAME -2 ALLOT

70 C, 60 C,
80 C, ( DL11 )
00 C,
```

00 C,  
42 C, BC40 ,  
80 C, ( DLI2 )  
00 C,  
40 C,  
0404 , 0404 , 0404 , 0404 ,  
0404 , 0404 , 0404 , 0404 ,  
0404 , 0404 , 0404 ,  
10 C,  
80 C, ( DLI3 )  
10 C,  
42 C, FOOTERMEM ,  
80 C, ( DLI4 )  
00 C,  
41 C, DLGAME ,

: SETDLGAME  
DLGAME SETDL ;

0 VARIABLE DLHELP -2 ALLOT

70 C, 60 C,  
80 C, ( DLI1 )  
00 C,  
42 C, BC40 ,  
80 C, ( DLI2 )  
00 C,  
00 C,  
40 C,  
40 C,  
0202 , 0202 , 0202 , 0202 ,  
0202 , 0202 , 0202 , 0202 ,  
0202 , 0202 , 0202 ,  
A0 C, ( DLI3 )  
00 C, 00 C,  
42 C, FOOTERMEM ,  
80 C, ( DLI4 ) 00 C,  
41 C, DLHELP ,

: SETDLHELP  
DLHELP SETDL ;

0 VARIABLE DLTITLE -2 ALLOT

70 C, 40 C,  
80 C, 00 , 42 C, BC40 ,  
80 C, 00 C, 00 C, 30 C,  
( 1) 4F C, 8010 , 0F C,  
0F0F , 0F0F , 0F0F  
( 2) 0F0F , 0F0F , 0F0F , 0F0F ,  
( 3) 0F0F , 0F0F , 0F0F , 0F0F ,  
( 4) 0F0F , 0F0F , 0F0F , 0F0F ,  
( 5) 0F0F , 0F0F , 0F0F , 0F0F ,  
( 6) 0F0F , 0F0F , 0F0F , 0F0F ,  
( 7) 0F0F , 0F0F , 0F0F , 0F0F ,  
( 8) 0F0F , 0F0F , 0F0F , 0F0F ,  
( 9) 0F0F , 0F0F , 0F0F , 0F0F ,  
( 10) 0F0F , 0F0F , 0F0F , 0F0F ,  
( 11) 0F0F , 0F0F , 0F0F , 0F0F ,  
( 12) 0F0F , 0F0F , 0F0F , 0F0F ,  
( 13) 0F0F , 0F0F , 0F0F , 0F0F ,

```
4F C, 9000 , 0F C,  
( 14) 0F0F , 0F0F , 0F0F ,  
( 15) 0F0F , 0F0F , 0F0F , 0F0F ,  
( 16) 0F0F , 0F0F , 0F0F , 0F0F ,  
( 17) 0F0F , 0F0F , 0F0F , 0F0F ,  
( 18) 0F0F , 0F0F , 0F0F , 0F0F ,  
( 19) 0F0F , 0F0F , 0F0F , 0F0F ,  
( 20) 0F0F , 0F0F , 0F0F , 0F0F ,  
( 21) 0F0F , 0F0F , 0F0F , 0F0F ,  
( 22) 0F0F , 0F0F , 0F0F , 0F0F ,  
( 23) 0F0F , 0F0F , 0F0F ,  
A0 C, 00 C, 00 C,  
42 C, FOOTERMEM ,  
80 C, 00 C, 41 C, DLTITLE ,
```

```
: SETDLTITLE  
DLTITLE SETDL ;
```

```
( Displaylist Interrupt )
```

```
D40A CONSTANT WSYNC  
D016 CONSTANT COLPF0  
D017 CONSTANT COLPF1  
D018 CONSTANT COLPF2  
D019 CONSTANT COLPF3  
D01A CONSTANT COLBK  
D409 CONSTANT CHBASE  
02F4 CONSTANT CHBAS
```

```
80 4 - VARIABLE FONT
```

```
0E VARIABLE WHITE  
00 VARIABLE BLACK  
06 VARIABLE GREY  
B6 VARIABLE GREEN  
84 VARIABLE BLUE
```

```
32 VARIABLE T-RED  
B2 VARIABLE T-GREEN  
82 VARIABLE T-BLUE
```

```
( Poor Mens Assembler )
```

```
48 CONSTANT PHA  
68 CONSTANT PLA  
40 CONSTANT RTI  
8D CONSTANT STA  
AD CONSTANT LDA  
A9 CONSTANT LDA#  
60 CONSTANT RTS  
20 CONSTANT JSR  
4C CONSTANT JMP
```

```
00 VARIABLE WHITEBAR -2 ALLOT
```

```
LDA C, WHITE ,  
STA C, WSYNC ,  
STA C, COLBK ,  
LDA C, BLACK ,  
STA C, WSYNC ,  
STA C, COLBK ,  
RTS C,
```



00 VARIABLE DLI1 -2 ALLOT

```
PHA C,  
JSR C, WHITEBAR ,  
STA C, COLPF2 ,  
LDA C, WHITE ,  
STA C, COLPF1 ,  
LDA# C, HERE DLI1 100 / C,  
STA C, VSDLST 1+ ,  
LDA# C, HERE DLI1 100 MOD C,  
STA C, VSDLST ,  
PLA C,  
RTI ,
```

." DLI1 is at " DLI1 . CR

00 VARIABLE DLI2 -2 ALLOT

( Patch DLI1 )

```
DLI2 100 MOD SWAP C!  
DLI2 100 / SWAP C!
```

```
PHA C,  
JSR C, WHITEBAR ,  
LDA C, GREY ,  
STA C, COLPF0 ,  
LDA C, WHITE ,  
STA C, COLPF1 ,  
LDA C, BLUE ,  
STA C, COLPF2 ,  
LDA C, GREEN ,  
STA C, COLPF3 ,  
LDA C, FONT ,  
STA C, CHBASE ,  
LDA# C, HERE DLI2 100 / C,  
STA C, VSDLST 1+ ,  
LDA# C, HERE DLI2 100 MOD C,  
STA C, VSDLST ,  
PLA C,  
RTI C,
```

." DLI2 is at " DLI2 . CR

00 VARIABLE DLI3 -2 ALLOT

( Patch DLI2 )

```
DLI3 100 MOD SWAP C!  
DLI3 100 / SWAP C!
```

```
PHA C,  
JSR C, WHITEBAR ,  
STA C, COLPF2 ,  
LDA C, WHITE ,  
STA C, COLPF1 ,  
LDA C, CHBAS ,  
STA C, CHBASE ,  
LDA# C, HERE DLI3 100 / C,  
STA C, VSDLST 1+ ,  
LDA# C, HERE DLI3 100 MOD C,  
STA C, VSDLST ,  
PLA C,  
RTI C,
```

." DLI3 is at " DLI3 . CR

00 VARIABLE DLI4 -2 ALLOT

( Patch DLI3 )

DLI4 100 MOD SWAP C!

DLI4 100 / SWAP C!

PHA C,

JSR C, WHITEBAR ,

LDA# C, DLI1 100 / C,

STA C, VSDLST 1+ ,

LDA# C, DLI1 100 MOD C,

STA C, VSDLST ,

PLA C,

RTI C,

." DLI4 is at " DLI4 . CR

E462 CONSTANT XITVBV

E45F CONSTANT SYSVBV

0224 CONSTANT VVBLKD

0222 CONSTANT VVBLKI

00 VARIABLE VBI -2 ALLOT

PHA C,

LDA# C, DLI1 100 / C,

STA C, VSDLST 1+ ,

LDA# C, DLI1 100 MOD C,

STA C, VSDLST ,

PLA C,

JMP C, SYSVBV ,

." VBI is at " VBI . CR

( Musik )

A001 CONSTANT MSTART

A008 CONSTANT MSTOP

A011 CONSTANT MINIT

A017 CONSTANT MPLAY

00 VARIABLE MUSIC

: SETMUSIC

MSTART @ 9F20 = IF

MINIT CALL

MSTART CALL

1 MUSIC !

THEN ;

: STOPMUSIC

MSTART @ 9F20 = IF

MSTOP CALL

0 MUSIC !

THEN ;

: STARTMUSIC

MSTART @ 9F20 = IF

MSTART CALL

1 MUSIC !

THEN ;

: SETVBI

NMIEN C@

```

0 NMIEN C!
VBI VVBLKI !
NMIEN C! ;

: SETDLI ( addr - )
( STOPMUSIC SETVBI ) VSDLST !
( STARTMUSIC ) ;

: RESETDLI 60 NMIEN C! COCE SETDLI ;

( Player Missile Grafics )

022F CONSTANT SDMCTL
026F CONSTANT GPRIOR
D000 CONSTANT HPOS0
D01D CONSTANT GRCTL
D407 CONSTANT PMBASE
D01C CONSTANT VDELAY
02C0 CONSTANT PCOLR0

: INITPM
SDMCTL C@ 8 + SDMCTL C! ( Enable PM DMA )
2 GRCTL C! ( Enable PM )
1 GPRIOR C!
80 6 - PMBASE C! ;

( Player )
0 VARIABLE CROSS -2 ALLOT
2 BASE !
00000000 C,
11101110 C,
10000010 C,
10000010 C,
00000000 C,
10000010 C,
10000010 C,
11101110 C,
00000000 C,

HEX

: PMPOS ( x y -- )
7A00 FF 00 FILL
YOffset @ 40 / +
8 * 7A0B +
LevelHeight @ 8 = IF 4 + THEN ( Hack! )
CROSS SWAP 9 CMOVE
XOffset @ 2 / +
8 * 34 + HPOS0 C!
10 VDELAY C! ;

( Strings )

: [ ] ( -- addr len )
R COUNT DUP 1+ R> + >R ;
: " ( -- addr len )
22 STATE @ IF
COMPILE [ ] WORD C@ 1+ ALLOT
ELSE

```

```
WORD DUP C@ 1+ PAD
SWAP CMOVE PAD COUNT
THEN ; IMMEDIATE
```

```
( Level )
```

```
: LEVEL1
```

```
7 LevelWidth ! 7 LevelHeight !
0 BloxUsed ! 1 PuzzleNumber !
```

```
BEGINLEVEL
```

```
PF" XXXXXXXX"
PF" XOO OOX"
PF" XO O OX"
PF" X O O X"
PF" XO O OX"
PF" XOO OOX"
PF" XXXXXXXX"
```

```
ENDLEVEL
```

```
;
```

```
: LEVEL2
```

```
A LevelWidth ! A LevelHeight !
0 BloxUsed ! 2 PuzzleNumber !
```

```
BEGINLEVEL
```

```
PF" XXXXXXXXXXXX"
PF" XO O OO OX"
PF" XO O OO OX"
PF" XOOO OX"
PF" XOOO OO OX"
PF" XO O OO OX"
PF" XO O OO X"
PF" X OO OX"
PF" X OO OX"
PF" XXXXXXXXXXXX"
```

```
ENDLEVEL ;
```

```
: LEVEL3
```

```
8 LevelWidth ! A LevelHeight !
0 BloxUsed ! 3 PuzzleNumber !
```

```
BEGINLEVEL
```

```
PF" XXXXXXXX"
PF" XO OOO X"
PF" X OOO X"
PF" XOO O X"
PF" X O OO X"
PF" X O OO X"
PF" XOO OOX"
PF" XOOO O X"
PF" X O OOX"
PF" XXXXXXXX"
```

```
ENDLEVEL ;
```

```
: LEVEL4
```

```
A LevelWidth ! A LevelHeight !
0 BloxUsed ! 4 PuzzleNumber !
```

```
BEGINLEVEL
```

```
PF" XXXXXXXXXXXX"
PF" X OOX"
PF" X OOOX"
```

```
PF" X 000000X"
PF" X0000000 X"
PF" X      OOX"
PF" X    000 X"
PF" X 000  X"
PF" X000  X"
PF" XXXXXXXXXXX"
ENDLEVEL ;
```

: LEVEL5

```
A LevelWidth ! A LevelHeight !
0 BloxUsed ! 5 PuzzleNumber !
```

BEGINLEVEL

```
PF" XXXXXXXXXXX"
PF" X0000  X"
PF" X000  X"
PF" X00    OX"
PF" X0000  OX"
PF" X000  OX"
PF" X00   000X"
PF" XO  00000X"
PF" X  000000X"
PF" XXXXXXXXXXX"
ENDLEVEL ;
```

: LEVEL6

```
A LevelWidth ! A LevelHeight !
0 BloxUsed ! 6 PuzzleNumber !
```

BEGINLEVEL

```
PF" XXXXXXXXXXX"
PF" X   00  X"
PF" X   00  X"
PF" X   00  X"
PF" X  000  X"
PF" X  000  X"
PF" X 0000  X"
PF" X0000  OOX"
PF" X000  OOX"
PF" XXXXXXXXXXX"
ENDLEVEL ;
```

: LEVEL7

```
A LevelWidth ! A LevelHeight !
0 BloxUsed ! 7 PuzzleNumber !
```

BEGINLEVEL

```
PF" XXXXXXXXXXX"
PF" X0000000 X"
PF" X000  OX"
PF" X000  OOX"
PF" X00 00000X"
PF" XO  0000X"
PF" XO  0000X"
PF" X00 00000X"
PF" X00 00000X"
PF" XXXXXXXXXXX"
ENDLEVEL ;
```

: LEVEL8

```
A LevelWidth ! 6 LevelHeight !
```

```
0 BloxUsed ! 8 PuzzleNumber !
BEGINLEVEL
PF" XXXXXXXXXXXX"
PF" XO O O OOX"
PF" XO OOO O X"
PF" XOO OO X"
PF" XOOO OOO X"
PF" XXXXXXXXXXXX"
ENDLEVEL ;
```

: LEVEL9

```
A LevelWidth ! A LevelHeight !
0 BloxUsed ! 9 PuzzleNumber !
BEGINLEVEL
PF" XXXXXXXXXXXX"
PF" XOO O OO X"
PF" XOOO O X"
PF" XOOOOO OOX"
PF" XO OX"
PF" XOO OOOX"
PF" X O X"
PF" XOOOOO OOX"
PF" X OOX"
PF" XXXXXXXXXXXX"
ENDLEVEL ;
```

: LEVELA

```
A LevelWidth ! 8 LevelHeight !
0 BloxUsed ! A PuzzleNumber !
BEGINLEVEL
PF" XXXXXXXXXXXX"
PF" X OOOOOO X"
PF" X O OOX"
PF" X OOOO O X"
PF" X O O O X"
PF" X O OOOOX"
PF" X OOOOO X"
PF" XXXXXXXXXXXX"
ENDLEVEL ;
```

: LEVELB

```
9 LevelWidth ! A LevelHeight !
0 BloxUsed ! B PuzzleNumber !
BEGINLEVEL
PF" XXXXXXXXXXXX"
PF" XOO OOX"
PF" XOO OOX"
PF" XOOO OOOX"
PF" XOOO OOOX"
PF" XOO O OOX"
PF" XOO OOX"
PF" XOO OOX"
PF" XOO OOX"
PF" XXXXXXXXXXXX"
ENDLEVEL ;
```

: LEVELC

```
9 LevelWidth ! 8 LevelHeight !
0 BloxUsed ! C PuzzleNumber !
```

```
BEGINLEVEL
PF" XXXXXXXXXXXX"
PF" X OO OO X"
PF" XO      OX"
PF" X  O O  X"
PF" X  O O  X"
PF" XO  O  OX"
PF" X  O O  X"
PF" XXXXXXXXXXXX"
ENDLEVEL ;
```

```
: LEVELD
A LevelWidth ! A LevelHeight !
0 BloxUsed ! D PuzzleNumber !
BEGINLEVEL
PF" XXXXXXXXXXXX"
PF" XOO      OOX"
PF" XO      OOOX"
PF" X      OOOOX"
PF" X      OOOX"
PF" X  O  OO X"
PF" X OOO  OX"
PF" XOOOOO  X"
PF" XOOOOOO  X"
PF" XXXXXXXXXXXX"
ENDLEVEL ;
```

```
: LEVELE
A LevelWidth ! A LevelHeight !
0 BloxUsed ! E PuzzleNumber !
BEGINLEVEL
PF" XXXXXXXXXXXX"
PF" XO OOOO OX"
PF" X O OO O X"
PF" XO O  O OX"
PF" X OOOOOO X"
PF" X  OOOO  X"
PF" X OOOOOO X"
PF" XO  OO  OX"
PF" XOOO  OOOX"
PF" XXXXXXXXXXXX"
ENDLEVEL ;
```

```
: LEVELF
A LevelWidth ! 7 LevelHeight !
0 BloxUsed ! F PuzzleNumber !
BEGINLEVEL
PF" XXXXXXXXXXXX"
PF" X  OOOOOX"
PF" XOO O  O X"
PF" X  OOO  X"
PF" X OO O OOX"
PF" XOOO O  X"
PF" XXXXXXXXXXXX"
ENDLEVEL ;
```

00 VARIABLE LEVELMEM 60 ALLOT
." LevelMem is at " LEVELMEM . CR

```
LEVELMEM 60 ERASE
00 VARIABLE MAXLEVEL
```

```
: LOADLEVEL
." Load Level Savefile..." CR
" D:STROQ.SAV" R/O OPEN-FILE DUP
IF ." Open Savefile Error:" . CR WAITKEY ELSE
DROP DUP LEVELMEM 60 ROT READ-FILE DUP
IF DROP DROP ELSE
DROP DROP CLOSE-FILE DROP ." ok" CR THEN THEN ;
```

```
: SAVELEVEL
." Store Level Savefile..." CR
" D:STROQ.SAV" W/O OPEN-FILE DUP
IF ." Open Savefile Error:" . CR ELSE
DROP DUP LEVELMEM 60 ROT WRITE-FILE DUP
IF ." Save Savefile Error:" . CR ELSE
DROP CLOSE-FILE DROP ." ok" CR THEN THEN ;
```

```
: LEVELADDR LEVELMEM @ 4 * LEVELMEM + ;
```

```
: >LEVEL ( xt -- )
1 LEVELMEM +! LEVELADDR ! ;
```

```
: GETLEVELXT ( -- xt )
LEVELADDR @ CFA ;
```

```
: NEXTLEVEL
1 LEVELMEM +!
MAXLEVEL @ LEVELMEM @ < IF
1 LEVELMEM !
THEN
GETLEVELXT EXECUTE ;
```

```
' LEVEL1 >LEVEL
' LEVEL2 >LEVEL
' LEVEL3 >LEVEL
' LEVEL4 >LEVEL
' LEVEL5 >LEVEL
' LEVEL6 >LEVEL
' LEVEL7 >LEVEL
' LEVEL8 >LEVEL
' LEVEL9 >LEVEL
' LEVELA >LEVEL
' LEVELB >LEVEL
' LEVELC >LEVEL
' LEVELD >LEVEL
' LEVEL E >LEVEL
' LEVELF >LEVEL
```

```
LEVELMEM @ MAXLEVEL !
```

```
: SAVESCORE
BloxUsed @ LEVELADDR 2+ ! ;
```

```
( Footer )
```

```
: CLEARFOOTER
FOOTERMEM 28 ERASE ;
```

```
: >FOOTER
BC40 FOOTERMEM 28 CMOVE ;
```

```
: FOOTER
```



```

CLEARFOOTER
0 0 AT ." Main Menu:Solve Puzzle:Reset Puzzle:Help"
>FOOTER ;
: TFOOTER
CLEARFOOTER
0 0 AT ." Start Game : Select Puzzle : Quit : Help"
>FOOTER ;

00 VARIABLE I1
00 VARIABLE I2

: FINV
I2 @ I1 @ DO
FOOTERMEM I + DUP C@ 80 XOR SWAP C!
LOOP ;

00 VARIABLE TSEL

: TMENU
BEGIN
TSEL @
DUP 0= IF 00 I1 ! 0B I2 ! THEN
DUP 1 = IF 0C I1 ! 1B I2 ! THEN
DUP 2 = IF 1C I1 ! 22 I2 ! THEN
3 = IF 23 I1 ! 28 I2 ! THEN
FINV KEY FINV
DUP 2B = IF TSEL @ IF -1 TSEL +! THEN THEN
DUP 2A = IF TSEL @ 3 < IF 1 TSEL +! THEN THEN
9B = UNTIL ;

00 VARIABLE GSEL
: GMENU
BEGIN
GSEL @
DUP 0= IF 00 I1 ! 09 I2 ! THEN
DUP 1 = IF 0A I1 ! 16 I2 ! THEN
DUP 2 = IF 17 I1 ! 23 I2 ! THEN
3 = IF 24 I1 ! 28 I2 ! THEN
FINV KEY FINV
DUP 2B = IF GSEL @ IF -1 GSEL +! THEN THEN
DUP 2A = IF GSEL @ 3 < IF 1 GSEL +! THEN THEN
9B = UNTIL ;

( Picture )
8010 CONSTANT PICBASE

: LOADPIC
" D:STROQ.PIC" R/O OPEN-FILE DUP
IF ." Pic Open Error " . CR BYE ELSE
DROP DUP PICBASE 1C20 ROT READ-FILE DUP
IF ." Pic Load Error " . . CR BYE ELSE
DROP DROP CLOSE-FILE DROP THEN THEN ;

( Font )

8000 400 - CONSTANT FONTBASE
0 VARIABLE SAVEFONT

: LOADFONT

```

```

" D:STROQ.FNT" R/O OPEN-FILE DUP
IF ." Font Open Error " . CR BYE ELSE
DROP DUP FONTBASE 400 ROT READ-FILE DUP
IF ." Font Load Error " . . CR BYE ELSE
DROP DROP CLOSE-FILE DROP THEN THEN ;

: SETFONT
CHBAS C@ SAVEFONT C!
FONTBASE 100 / CHBAS C! ;

: RESETFONT
SAVEFONT C@ CHBAS C! ;

: CROSSPOS
curx @ cury @ PMPOS ;

: CURUP
cury @ IF -1 cury +! THEN ;
: CURDOWN
cury @ LevelHeight @ 1 - < IF 1 cury +! THEN ;
: CURLEFT
curx @ IF -1 curx +! THEN ;
: CURRIGHT
curx @ LevelWidth @ 1 - < IF 1 curx +! THEN ;

00 VARIABLE Pulse -2 ALLOT
00 C, 00 C, 00 C, 00 C,
02 C, 02 C, 02 C, 02 C,
04 C, 04 C, 04 C,
06 C, 06 C,
08 C,
0A C,
0C C,
0A C,
08 C,
06 C, 06 C,
04 C, 04 C, 04 C,
02 C, 02 C, 02 C, 02 C,
00 C, 00 C, 00 C, 00 C, 00 C,

00 VARIABLE PulseP
00 VARIABLE Count

: WAIT 0 14 C! BEGIN 14 C@ UNTIL ;
: NOCLICK FF 2DB C! ;

: RESET
GETLEVELXT EXECUTE 0 STROQPTR !
CLS PRINTLEVEL HEADER ;

: DOS
0 HPOS0 C! CLS RESTORED L CURON STOPMUSIC
SAVELEVEL BYE ;

: SETGAME
80 4 - FONT C! 82 BLUE !
DLI1 SETDLI SETDLGAME
FOOTER CLS HEADER PRINTLEVEL ;

```

```

: SETTITLE
  0 BLUE !
  TFOOTER CLS HEADER
  SETDLTITLE DLI1 SETDLI ;

: HELP
  RESETDLI CLS 0 HPOS0 C!
  CHBAS C@ FONT C! B2 BLUE !
  SETDLHELP DLI1 SETDLI
  0 0 AT ."          Stroq Help "
  0 2 AT ." Atari Version
  0 3 AT ." (C) 2005 Carsten Strotmann &"
  0 4 AT ."          Winfried Piegsda"
  0 6 AT ." Stroq is OpenSource Software"
  0 7 AT ."          and Licensed under the GPL"
  0 8 AT ."          Version 2 or higher"
  0 A AT ." How to play:"
  0 B AT ." -----"
  0 D AT ." The goal of the game is simple, draw"
  0 E AT ." a single continuous line, the stroke"
  0 F AT ." from one square to another."
  0 10 AT ." When you run the puzzle (Solve Puzzle)"
  0 11 AT ." green and blue squares along the line"
  0 12 AT ." will flop over (blue becoming green,"
  0 13 AT ." green becoming blue). The Puzzle is"
  0 14 AT ." solved when all tiles on the row are"
  0 15 AT ." the same color."
  WAITKEY CLS
  RESETDLI
  ;

: MOVECURSOR
  3 GSEL !
  BEGIN
  ?TERMINAL
  IF KEY
    DUP 1C = IF CURUP      THEN
    DUP 2D = IF CURUP      THEN
    DUP 1D = IF CURDOWN    THEN
    DUP 3D = IF CURDOWN    THEN
    DUP 1E = IF CURLEFT    THEN
    DUP 2B = IF CURLEFT    THEN
    DUP 1F = IF CURRIGHT   THEN
    DUP 2A = IF CURRIGHT   THEN
    DUP 9B = IF SETBLOX HEADER THEN
    DUP 20 = IF SETBLOX HEADER THEN
    DUP 08 = IF HELP SETGAME THEN
    DUP 11 = IF DOS        THEN
    DUP 12 = IF RESET      THEN
    DUP 13 = IF SOLVE
              IF SAVESCORE NEXTLEVEL
              THEN
              RESET THEN
    DUP 1B = IF GMENU
              GSEL @
              DUP 1 = IF SOLVE IF
                      SAVESCORE NEXTLEVEL THEN
                      RESET THEN
              DUP 2 = IF RESET THEN

```

```

                3 = IF HELP SETGAME THEN
                THEN
    DUP 0D = IF MUSIC @ IF STOPMUSIC
                ELSE STARTMUSIC
                THEN
                THEN
    DROP ( 0 0 AT ." SP=" SP@ . )
    CROSSPOS
ELSE
    WAIT 1 Count +! Count @ 40 AND 0=
    IF
        PulseP @ 1+ 2F AND PulseP !
        Pulse PulseP @ + @ PCOLR0 C!
    THEN
    THEN
GSEL @ 0 = UNTIL 0 HPOS0 C!
RESETDLI ;

: SELECTPUZZLE
RESETDLI CLS 0 HPOS0 C!
CHBAS C@ FONT C! 32 BLUE !
SETDLHELP DLI1 SETDLI
C 0 AT ." Select Puzzle "
DECIMAL
BEGIN
    2 5 AT ." Level:" LEVELMEM @
    S>D <# # # # #> TYPE SPACE
    2 7 AT
    LEVELADDR 2 + @
    DUP IF ." Solved with "
        S>D <# # # # #> TYPE SPACE
        ." Blox used."
    ELSE
        DROP
        ." Not solved.          "
    THEN
    KEY
    DUP 2B = IF LEVELMEM @ 1 >
                IF -1 LEVELMEM +!
                THEN
                THEN
    DUP 2A = IF MAXLEVEL @
                LEVELMEM @ > IF
                1 LEVELMEM +!
                THEN
                THEN
    9B = UNTIL HEX
    GETLEVELXT EXECUTE
    CLS RESETDLI ;

: TITLE
SETTITLE
TMENU
84 BLUE ! RESETDLI ;

: STROQ
CLS SAVEDL
0 2C6 C! NOCLICK
." Starting Stroq..." CR

```

```
( ." Load Font..." CR )
( LOADFONT )
( ." Load Picture..." CR )
( LOADPIC )
LOADLEVEL
0 LEVELMEM !
NEXTLEVEL
INITPM SETMUSIC
BEGIN
  TITLE
  TSEL @
  DUP 0= IF
    SETGAME
    CROSSPOS MOVECURSOR
  THEN
  DUP 1 = IF SELECTPUZZLE THEN
  DUP 2 = IF DOS THEN
    3 = IF HELP THEN
  AGAIN
;

." Ready." CR
```

## Downloads#

The Atari Stroq Programm has some quirks when running in an Emulator (such as [Atari800](#)). It runs fine on the real hardware.