



# VOLLEYRING

Sonic vs Shadow

EXTREME-256 BASIC game for MSX

(c) 2021 Nick Shcherbyna (Lviv, Ukraine)      computer.history.lviv@gmail.com

## Instructions /

Hedgehogs cannot play volleyball because their spines might pierce the ball. Instead they choose volleyring - essentially the same game, but using a ring from Sonic's universe. The rules are simple: send a ring to opponent's side, don't touch the net or ground. Hedgehog have to jump to strike the ring, but three consecutive contacts with it is a fault. Match is played to 15 points.

Left player is **Sonic the Hedgehog**, right player is **Shadow the Hedgehog**. Both are controlled by players using joysticks or cursor keys. Press **[ 1 ]** in the main menu to choose Sonic's controls, press **[ 2 ]** to pick Shadow's controls. Press **[ S ]** to start match. Note that there is an easter egg in the game!

The game is written in **MSX BASIC**, and can run even on **MSX1**. For crazy action pick **MSX Turbo R**. If you do not have real hardware, use emulator like **WebMSX** (<https://webmsx.org/>).

### General tips:

For single-player action choose the same controls for Sonic and Shadow. Then you will control both hedgehogs simultaneously. Playing with someone else is likely more enjoyable, but this mode is good for practice.

### WebMSX tips:

Copy BASIC source to clipboard, then press **[ Alt ]-[ V ]** and **[ Ctrl ]-[ V ]**, after that press **[ F5 ]** to run the game.

Press **[ Alt ]-[ T ]** to change the CPU frequency (and game speed).

Press **[ Alt ]-[ K ]** several times to activate double joysticks emulation via keyboard. Note that the first joystick is mapped to cursor keys, and the 2nd one is mapped to **[ F ]** (left), **[ T ]** (jump), and **[ H ]** (right).

Alternatively, one can run game from disk, either automatically or manually:  
`LOAD "VOLLEY10.BAS",R`

#### Disk Contents /

`AUTOEXEC.BAS` - Autoloader  
`VOLLEY10.SC2` - Splash screen  
`README.TXT` - This document  
`VOLLEY10.BAS` - 10-Liner game

#### Source Code Explained /

##### Variables:

`(U, Y + W)` - Sonic's coordinates. `Y` is constant, `W` becomes negative on jump, otherwise 0.  
`(X, Y + Z)` - Shadow's coordinates, like above.  
`(P, Q)` - Ring coordinates.  
`(R, S)` - Ring speed vector. Mirrored on screen bounds, updated when hedgehog strikes the ring in jump. `V` is used to simulate physics - every 128 steps `|R|` is decreasing, and every 8 steps `S` is increasing.  
`N` - Number of consecutive contacts. Reset to 0 when the ring passes the net.  
`T` - Temporary variable, mostly used for joystick and keyboard readings.  
`M(2)` - Array of scores per player.  
`K(2)` - Array of controls per player (i.e., `STICK()` arguments).  
`A(3)` - Array of control names (cursor keys, joystick 1/2).

##### Functions:

`FNM(V)` - Mirroring bits in byte `V`. Used to generate Shadow's sprite from Sonic in three steps (C code):  
    `V = ((V >> 1) & 0x55) | ((V & 0x55) << 1); // 1) Swap bits`  
    `V = ((V >> 2) & 0x33) | ((V & 0x33) << 2); // 2) Swap pairs`  
    `V = ((V >> 4) & 0x0F) | ((V & 0x0F) << 4); // 3) Swap nibbles`  
`FNR(V,S,M)` - Helper function for `FNM(V)`, performs single step from above.  
`FNK(J)` - Wrapper for `STICK()` for player `J`.  
`FNJ(J)` - Helper for jumps. Assumes `T` is `STICK()` reading. Argument is `W` or `Z`, for Sonic or Shadow. Returns -1 if stick is up and jump is allowed, 0 otherwise.

##### Tricks:

As already stated above, Shadow's sprite is mirrored from Sonic. Also, first ten scanlines of two Sonic's sprites are similar, so there is no 2nd copy.  
Ring is generated from its quarter by mirroring horizontally and vertically.  
Net is created using `STRING$(10,"#")`, as the same pattern for all scanlines.  
Many IF statements are replaced by math, since MSX BASIC represents FALSE as 0, and TRUE as -1. An example of this technique is changing ring vector when it touches screen bounds: `R=R*SGN(1+2*(P<10RP>239))`. The result of `SGN` is -1 for bounds, 1 otherwise.

##### Lines:

0 Initialize graphics screen, define integer and string variable prefixes to save on % and \$, prepare control names, draw grass and sun.  
1 Define helper functions, define the ring sprite, prepare hedgehog sprites.  
2 Define hedgehog sprites, define and draw the net.  
3 Initialize core variables, draw checkered ground, finish grass, play intro music.  
4 Display game title, copyright, and main menu. Keyboard handling for menu.  
5 Control handling and sprite moving for Sonic, then do the same for Shadow. Move the ring by the current speed vector.  
6 Reset the consecutive contacts counter if needed. Handle Sonic's strike.  
7 Handle Shadow's strike.  
8 Simulate ring physics. Check for all faults, update score and variables if needed. Otherwise flip the speed vector, if the ring hits screen bounds.  
9 Subroutine that shows score and checks for victory. If so, puts the winner name and restarts the game.

#### Acknowledgements /

Thanks to my son Dan for the game idea, ad picture and sprites.  
Thanks to SEGA for Sonic and Shadow hedgehog characters.  
Thanks to Microsoft and ASCII Corp. for MSX BASIC and machines.  
Thanks to Arcade Volleyball authors for some inspiration.