

A strategic puzzle game for the 16K, 48K and 128K Sinclair ZX Spectrum

Written by [@Blerkotron](#)
(C)2017, Blerkotronic Software

Graphics very kindly created by [@hot_piping](#)

Find the latest version at
<https://blerkotron.itch.io/minesweeper>

Introduction

Minesweeper is a ZX Spectrum implementation of the classic Minesweeper puzzle, familiar to many people as one of the games bundled with Microsoft Windows.

This isn't the first version of the game for the Spectrum (and probably won't be the last) but as far as I can tell from the World of Spectrum database it does seem to be the first version to be playable on a 16K machine!

Loading the game

There are two different versions of Minesweeper – a 16K edition and an enhanced 48K version. 16K owners can only play the 16K version (obviously) but 48K and 128K users can choose to play either.

The 48K edition features more options (like the 'Heroes' mode where the boards are based on classic Speccy sprites) and some more polish on the front-end but the game is functionally the same regardless of which version you choose, so if you own a grown-up Speccy and just want to play classic Minesweeper then the 16K version is a tad faster to load.

16K/48K

Type LOAD "" and press ENTER, then play the tape.

128K/Plus 2/Plus 3

Use the LOADER function from the main menu, then play the tape.

The game will also work on your favourite Spectrum emulator, and has been tested as working (under emulation, at least) on the Pentagon 128 and the Timex series of machines.

The main menu

Various options are available from the main menu that let you customise your game. Press the key listed alongside the option to change it.

Size

Selects the type and size of game board that you want to use from a series of predefined sizes and a number of custom modes.

SMALL: standard 9x9 board
MEDIUM: standard 16x16 board
LARGE: standard 30x16 board
HEROES: boards based on classic Spectrum sprites! (48K version only)
CUSTOM: define your own board by customising height, width and number of mines

Level

The function of this option changes depending on the setting you selected for *Size*.

SMALL: select between easy (10 mines) or hard (35 mines)
MEDIUM: select between easy (40 mines) or hard (99 mines)
LARGE: select between easy (99 mines) or hard (170 mines)
HEROES: choose from one of twenty heroic board designs (48K version only)
CUSTOM: select your own height, width and number of mines

Display

Choose between various backdrops modes to brighten up your in-game display.

AMAZE: a randomly-generated maze-like backdrop to amaze and delight
PRETTY: a backdrop of aesthetically-pleasing hand-crafted tiles
PLAIN: if the other options make your eyes go weird you can turn backdrops off entirely

Control

Use this option to select keyboard or joystick control. Various popular joystick interfaces are supported.

The keys are:

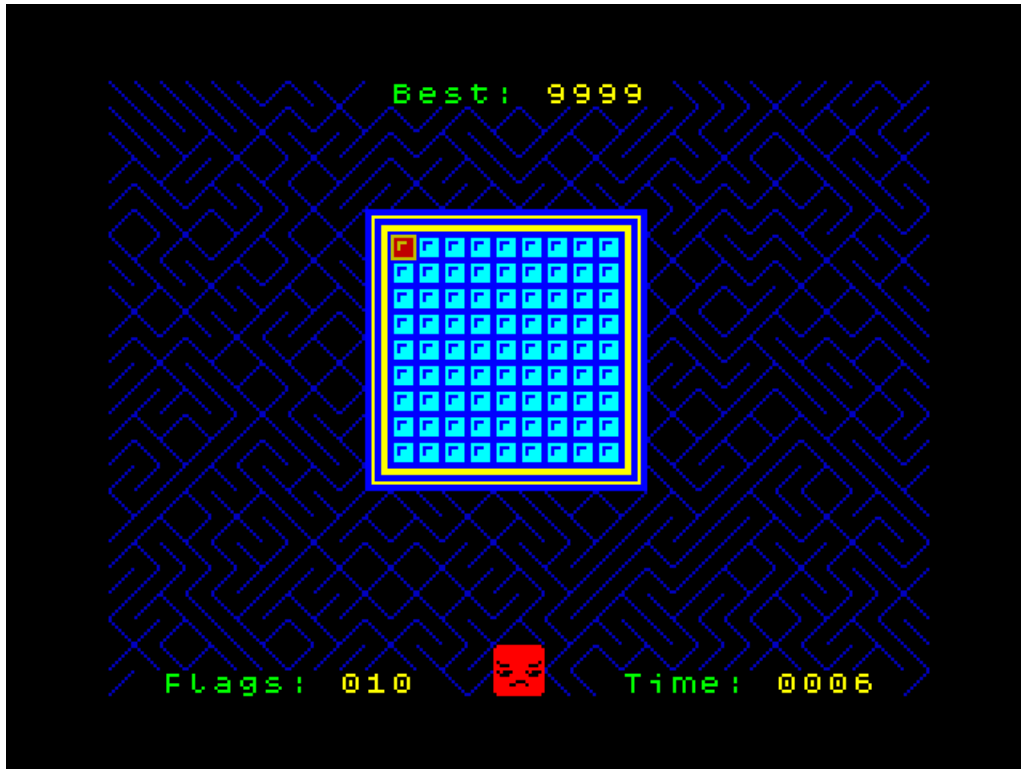
Q, *A*, *O* and *P* to move cursor
SPACE to activate a square
M to mark a square

Or use a joystick to move, the fire button to activate and *SPACE* to mark.

How to play Minesweeper

Minesweeper is a very simple concept but can be quite tricky to play. Basically you must find all of the mines in the grid without clicking on one and causing it to explode!

Upon starting the game the screen will display the board with all squares covered.



Bottom-left is the number of flags you have left to place – one to mark each mine hidden in the board. Bottom-right is the timer showing how long you have been playing – you can't run out of time, but the faster you find all the mines the more likely you are to get a high score. The current high score (one for each board/difficulty except for custom mode) is displayed at the top of the screen.

Your cursor is displayed in red and yellow and can be moved using the keyboard or joystick.

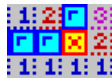
Activate a covered square to uncover it. If you uncover a mine then it's game over!



If the square doesn't contain a mine you will see a number indicating how many mines are present in the squares surrounding it. Use these as clues to calculate where the mines are hidden.



If you think you know where a mine is hidden you should mark it with a flag, which prevents you from accidentally uncovering it.

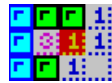


Clicking on an uncovered square has different effects depending on the state of the surrounding squares:

- If the square has the correct number of flags surrounding it then any covered squares will be uncovered.



- If the square doesn't have enough flags then any covered squares will flash green.



- If a square has too many flags then the flags will flash red to show that you have made a mistake.



If you manage to uncover all of the squares which *aren't* hiding a mine then you win and the game is over – and if you did it fast enough then you might be able to beat the high score for that board.

And that's basically it – if you need more help then there are dozens of sites on the Internet with tutorials and videos to help you along but really the best way to get into it is to just have a go.

Have fun!

Acknowledgements

Minesweeper came about because I decided it was finally time I sat down and tried to learn Z80 machine code. I chose Minesweeper because it's a relatively simple (but not too simple) concept and it seemed like the ideal way to try and flex some programming muscles that I had no idea if I actually had or not. As it turned out, it was *enormous* fun to write and I learned a hell of a lot about Z80 on the way – ten year old me would be in awe of (cough) somewhat-older me.

Along the way I've had help and advice from a number of excellent folk who I'd like to thank personally for their time and patience:

[@hot_piping](#) is both a super fellow and a superb artist. After helping out on my last game he very, very kindly assisted me again on the graphical front, so anything you see which isn't very basic is his rather than mine. If you want pixels, he is most definitely your man. Did I mention that he's super? He is.

[@SuminellStudios](#) suggested the inclusion of boards shaped like classic Spectrum heroes, a genius idea that was a lot of fun to implement. Huge thanks to them, and do check out their excellent Speccy-themed shooter, [Z-Exemplar!](#)

[@Valerian48k](#) suggested I could adapt a classic C64 BASIC 'maze generator' demo to create random backdrops for the game – another really fun mini project which eventually became the 'AMAZE' display mode.

[@reidrac](#) is a source of constant amazement to me and his programming talents on multiple retro platforms seem to know no bounds. I'd like to thank him for his Z80 help and advice throughout the project and strongly suggest you check out all of his games (over at [usebox.net](#)) because they're all brilliant. One day I hope to be able to make games at least half as good as him. :-)

[@BobsStuffGames](#) (aka Bob Smith) has produced some brilliant Spectrum and ZX81 games for a long time now and it was reading his '[8-bit blog](#)' Z80 tutorials which finally persuaded me to sit down and try this Z80 thing seriously. If you're interested in getting into programming yourself then that's a cool place to start, or you could just [download his games](#) instead!

[@ZXspectrumDev](#) (aka Jonathan Cauldwell) remains one of the Spectrum's most inventive developers, and his 'How to Write Spectrum Games' tutorial was a big help during development of this game. Grab a copy of the tutorial and all of his excellent games from [his web site!](#)

[@Quantum_Sheep](#) played an early beta of the game and offered some lovely feedback on look and feel (despite not knowing how to play Minesweeper) and gave me plenty of pep talks along the way. You should definitely, definitely [play his games too](#), because he's a lovely chap and also the world's biggest tea fan!

And to everyone else who commented on or retweeted my practically endless tirade of pictures and ramblings during the development of the game – thank you! Your interest genuinely kept me going to the end. :-)

Hugs!

[@Blerkotron](#), 26/11/17

Speccy 'Heroes' thanks

Minesweeper includes levels based on graphics from the following titles:

'Blob' from 'Starquake' by Stephen Crow, (C)1985 Bubble Bus Software

'Charlie Fotheringham-Grunes' from 'Nodes of Yesod' by Steve Wetherill, Colin Grunes, and Stuart James Fotheringham, (C)1985 Odin Computer Graphics

'Daley Thompson' from 'Daley Thompson's Decathlon' by Paul Owens, Christian Urquhart, and F. David Thorpe, (C)1984 Ocean Software

'Dizzy' from 'Treasure Island Dizzy' by The Oliver Twins and Neil Adamson, (C)1988 Code Masters

'Egghead' from 'Egghead 4: Egghead Entertains' by Jonathan Cauldwell, (C)2006 Jonathan Cauldwell

'Frank N Stein' from 'Frank N Stein' by Colin Stewart, (C)1984 PSS

'Fred' from 'Fred' by Carlos Granados Martinez, Paco Menendez, Fernando Rada Briega and Juan Delcan, (C)1984 Quicksilva

'Henhouse Harry' from 'Chuckie Egg' by Nigel Alderton, (C)1984 A n' F Software

'He and She' from 'Ant Attack' by Sandy White and Angela Sutherland, (C)1983 Quicksilva

'Horace' from 'Hungry Horace' by William Tang, (C)1982 Psion Software/Melbourne House

'Learic' from 'Feud' by John and Ste Pickford and Pete Harrison, (C)1987 Bulldog Software

'Maria and Willy' from 'Jet Set Willy' by Matthew Smith, (C)1984 Software Projects

'Magic Knight' from 'Stormbringer' by David Jones and Ray Owen, (C)1987 MAD Games

'Monty Mole' from 'Monty on the Run' by Peter Harrap, (C)1985 Gremlin Graphics

'Rockford' from 'Boulderdash' by Peter Liepa and Chris Gray, (C)1984 Front Runner

'Technician Ted' from 'Technician Ted' by Steve Marsden and David Cooke, (C)1984 Hewson Consultants

'Wally Week' from 'Everyone's a Wally' by Chris Hinsley and Neil Strudwick, (C)1985 Mikro-Gen

'Wizball' from 'Wizball' by Paul Owens, Mark R. Jones and Peter Clarke, (C)1987 Ocean Software

'Sabre Wulf' from 'Knight Lore' by Tim and Chris Stamper, (C)1984 Ultimate Play The Game

Tools and sources used in the creation of Minesweeper

Minesweeper came to life thanks to the efforts of a great many people who provided either tools, utilities or library code that helped me along.

Z80 development environment:

[Visual Studio Code](#) editor by Microsoft
[PASMO Z80 cross-assembler](#) by Julián Albo
[ZX-Paintbrush](#) graphic editor by Claus Jahn
[SpecEmu 3.1](#) ZX Spectrum emulator by Mark Woodmass
[Bitbucket](#) source control by Atlassian

Library routines and utilities:

[BeepFX](#) sound effects suite by Shiru
[ZX7](#) data compression suite by Einar Saukas

Reference material and tutorials:

[Z80 CPU User Manual](#) by Zilog
[Programming the Z80 \(3rd Edition\)](#) by Rodney Zaks
[Bob's 8-bit blog](#) by Bob Smith
[How to Write Spectrum Games](#) by Jonathan Cauldwell
<http://z80-heaven.wikidot.com/>
<http://www.z80.info/>
<http://www.cpcwiki.eu>
<http://map.grauw.nl/sources/external/z80bits.html>
http://wikiti.brandonw.net/index.php?title=Calculator_Documentation#Z80_programming
<http://www.breakintoprogram.co.uk/>

Many thanks to everyone – I couldn't have done this without you!

Revision history

1.0 (01/12/17)

- Mines are now seeded when you uncover your first cell rather than when the game starts. This means I can 'rig' the outcome of that first click to make sure you *never* hit a mine on your first go – a suggestion made by several people who played 0.99. It does mean a tiny pause after that first click on larger boards, but I think it's worth it (I will keep trying to boost the speed of mine-laying in later versions).
- Fixed issue where clicking an uncovered cell could sometimes accidentally reduce the flag counter (occasionally below zero)
- Custom game message is now yellow rather than magenta to make sure it stands out if the background is also magenta.
- Board drawing is a tiny bit faster.
- Added version number to main menu.

0.99 (26/11/17)

- First public beta release – feature complete, but missing the final loading screen and still has a few placeholder graphics which I've drawn (not very well) myself.