Gerald's Column
by Gerald Fitton

A couple of months ago I described an addictive game called Mine Sweeper which you can play using a Fireworkz or Resultz spreadsheet. I included a demo version of Resultz on the Archive monthly disc and elsewhere so that anyone who wished could play it.

About the game I said:

"One of the 'problems' with the original game is that it is almost impossible for anyone to complete the 20 by 20 square (400 cells containing 60 mines) and get it right. Of course practice helps and some people manage to get half way—but I know of nobody who has completed this game successfully! This version of the game is too hard."

The Solution

Two people, Anthony Hilton and Tonnie Demarteau, have written to me saying that they have successfully completed one or more games! A third, Colin Singleton of Puzzle Corner fame, has sent me a BASIC programme which not only plays the game automatically but also provides statistics about the proportion of wining games.

Modifications

Nobody has written to me modifying the game to a smaller board nor to allow a few non fatal encounters with the mines before being wiped out. I suspect that the difficulty is either understanding the custom function which does the work or lack of familiarity with unusual functions such as the 'doubleclick' function. Even if you are unable or unwilling to learn the custom function programming language there are some features of the game which deserve a mention if only because they have appeared in wish lists for Fireworkz.

Features

When you double click on one of the squares the content will change from a blank to either an upper case M or a number between 0 and 8. This feature is controlled by the little used 'doubleclick' function.

If the number is zero there can not be any mines in adjacent squares so Resultz 'turns over' all of the surrounding squares for you. This feature is achieved by the custom function.

I have often been asked how the contents of cells can be changed semi-automatically. The cells in the file [!PlayMe] are changed by the custom function as the game progresses.

If you study the custom function then you will see how these features are implemented.

Certainly if you decide that you would like to modify the game then you will soon see how the custom function, [c_game], achieves these interesting semi-automatic effects.

Winning

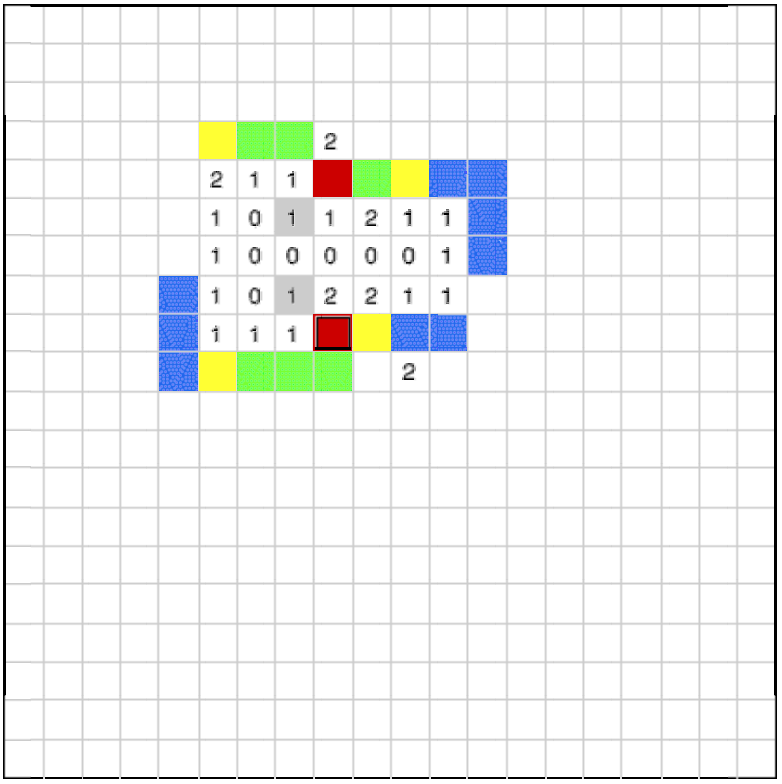
Now to strategies for winning the game.

Anthony

Anthony describes his technique:

"With a new board double-click at random hoping to get an area cleared by hitting a 0 square. Statistically if you reveal a 1 the probability of an adjacent square being clear is 87.5% and better than all the remaining squares which you know nothing about (85% at the start). But I rarely go with statistics that way.

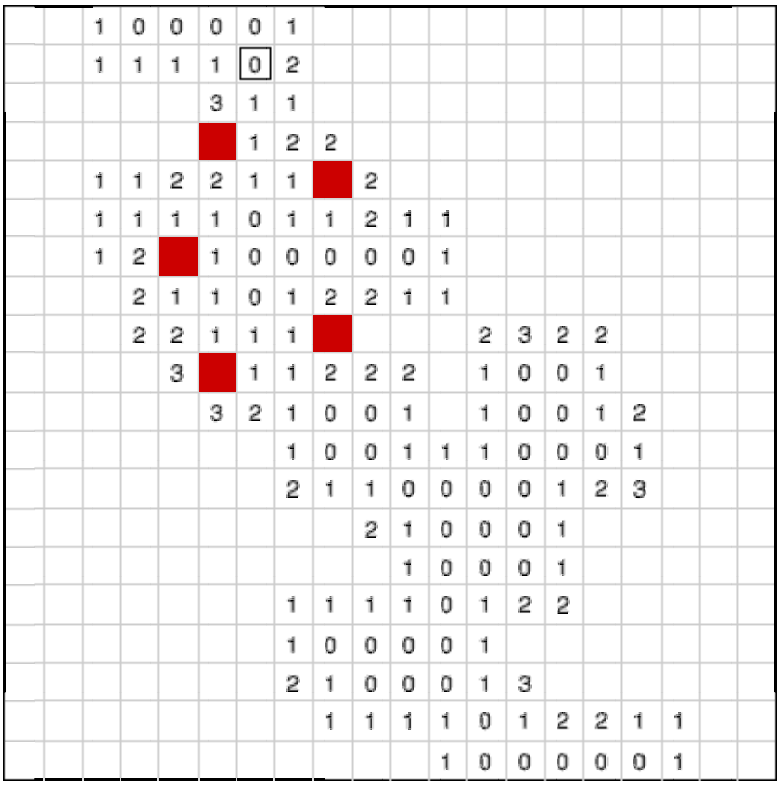
"In the screenshot below I first got the 2 which is on its own, then a 0 so all the other numbers were revealed."



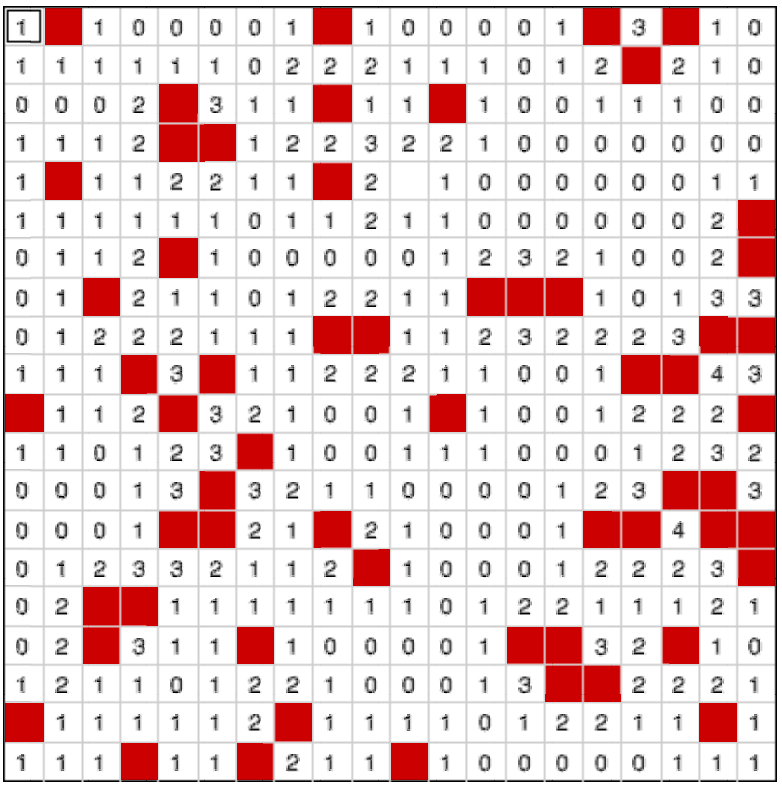
"Now look for any numbers which define where mines are (the 1s marked in grey in the screenshot) and mark those mines in red (click once on the cell followed by <Ctrl F1>).

"This leaves a number of squares which cannot contain mines (green) so clear those. Now at least those I've made yellow contain mines so mark them and clear the blue ones. There should now be more which can be cleared and others which definitely contain mines.

"In the screenshot below I have cleared the squares I'd indicated on the previous screenshot which appear in green and marked some of those I'd indicated in yellow."



“Slow and careful progress yields results, eventually to the screenshot shown below where the top left square with the bold outline was the last to be cleared.”



Anthony forgot one square—can you work out what it contains?

Anthony concludes with:

"Often there will be a situation where a small area contains mines but there is no way of knowing which squares contain the mines. They are (a) two squares with one mine to find and either satisfies the indicated squares, or (b) a square of four squares which holds two mines and either diagonal pair will satisfy the criteria."

Tonnie

Tonnie sent me a similar solution. She says:

"Basically it is very simple. The patterns aren't in the table, but in my mind."

1 When a (suspected) bomb is surrounded by eight 1s, all the surrounding 16 cells can safely be clicked.

2 This is, of course also true for less cells with 1s, touching a (suspected) bomb.

3 When 2 (suspected) bombs are neighbours, all the cells connected to 2s and 1s can safely be clicked.

4 When there is a partly fill (done by the program), there are always 'empty' cells (containing not recognised bombs) surrounded by seven filled cells and one that safely can be clicked.

"In general, following these guidelines will lead to the outcome of finding all 60 mines. Of course, it needs good concentration, for one 'slip of the mind' is destructive."

About Anthony's solution Tonnie says: "Thanks for sending it to me. I have given it a closer look and have come to the conclusion that the approach of Anthony is along the same lines as mine. He also has to trust his luck in the beginning, waiting for a partial fill. He also decides what can be clicked safely, based on the numerals. He also concludes there are a few situations in which there are too few clues. He also concludes that the player has to calculate very carefully. A mistake is deadly like being in a real mine field!"

Colin

Many of you will know Colin Singleton's excellent Puzzle Corner in Archive. I am an ardent fan even though I rarely make any submissions!

Colin decided on a completely different approach. He decided not to try the Resultz version of the MineSweeper game but to use BASIC. He created (using BASIC) a version of the same game with 60 mines on a 20 by 20 board. Then he wrote BASIC code to play the game. I have included Colin's BASIC program on the monthly Archive disc and you will find it on the Archive web site as well as our own.

First I shall describe the algorithm which he uses and then I shall quote some of his results. Finally I shall describe how to use Colin's BASIC program.

Since there are 60 mines in a 400 square board the initial probability that any square contains a mine is 0.15 (or 15% if you prefer percentages). Of course the first move has to be a guess and 15% of games end immediately after the first move.

There is an advantage to choosing a corner square for the first move.

This advantage is not that the probability of a mine is less in a corner but it is because the probability of the corner square containing a zero (no adjacent mines) is higher than any other square; Colin has calculated that this probability is 61% compared with only 27% for a 'non edge' square. If you are lucky and hit a zero then more squares will be revealed.

Later in the game, if a square is found to have more than zero adjacent mines (this number can be between 1 and 8) and if you have defused all those mines then all the remaining neighbours of that square can be selected without risk.

Conversely if the number of adjacent mines is equal to the number of 'unknown' adjacent squares then all must contain mines!

Although it is possible that the board layout will be such that no risks need be taken after the first move, this is unlikely. Colin's program chooses a square at random in these circumstances. Colin suggests that his program could be improved because some squares are less risky than others. As a simple improvement I would suggest selecting another corner (if available) or an edge square—but a more advanced algorithm might indicate that some squares in the body of the board are less risky than others.

Colin ran his program 8000 times with the standard board (60 mines on a 20 by 20 board). He says that of these games 34% were successful. The accuracy of this sample (of 8000 games) is such that we can say, "A skilled player using Colin's algorithm can successfully complete the game about one third of the time!". That was a surprise to me!

Colin's BASIC program is included in two forms. The program [Mines_0001] runs a single game and [Mines_5000] runs 5000 games. He says that it is just something he dashed off and is not meant as an exhibition piece with neat and well commented coding! It is not a WIMP program but it does work well.

You can change the size of the board (up to 20 by 20 maximum) and the number of mines at line 190. You can run your own experiment changing the number of games at line 890. Turned squares are indicated by asterisks. The program lists the numbers and percentage of games which have resulted in the defusing of each possible number of mines.

The five figures at the bottom are:
The Number of games played
Number of games not completed successfully
Number of partly successful games
Total number of mines defused in partly successful games
The Average number of mines defused in partly successful games

Colin's final comment is that if the program is lucky enough to defuse one mine then it will defuse most of the mines. It would seem that if (as a human player) you are lucky with your first square then you can work through most of the rest of the mines most of the time.

Perhaps that is what makes it a good game? You stand a very good chance of winning (after the first move) if you are careful—this makes it a game of skill with only a small element of chance.

Colin can be contacted at:

Colin R J Singleton
41 St Quentin Drive
Sheffield
S17 4PN

Conclusion

I must now eat 'humble pie' or at least change my comment which was to the effect that the game in its standard form is too difficult for mere mortals. After all, it would seem that a skilled player either fails with the first mine through sheer bad luck or will work through the board systematically taking few additional chances.

If you run Colin's program a few thousand times and get some statistics out of it (for example a graph showing the percentage of games which result in 0 to 60 mines being defused) then please let me have your results for publication.

If you improve Colin's algorithm (and his BASIC program) then please let me know.

My 'Small Competition' to change the game by modifying the custom function is still running. I have received no replies at all. Is it that you need more information about the custom function and how it works?

Finally

You can write to me at the address given in Paul's Fact File. I do give help with spreadsheets including Eureka, Schema, Fireworkz and PipeDream. If you do need some help then please let me have an example file which you have created; it saves me so much work and reduces the chances of misunderstanding your problem. Return postage and a self addressed label will be appreciated.