

Confound Game

by

Pete Koziar

3602 Cedar Drive
Baltimore, MD 21207

1. Copyright Notice

This document and accompanying program are copyright 1993 by Pete Koziar. This means that you may:

1. Try this program out for a time, after which, if you like it, you must register it for the sum of \$15.00 American.
2. Freely share this program among friends, relatives, BBS's, etc. so long as all pieces are included, i.e., the program itself and all documentation, including this document itself.
3. Distribute this program on diskette, either by itself, or as part of a collection of programs, so long as you make it clear to the purchaser that it is not "freeware" nor "public domain," but "shareware," and that the price he paid for the diskette does not include the registration fee for this program. You also are not to charge an excessive (over \$15.00 in 1993 dollars) fee for the service.

You may not, however:

1. Modify any file in this collection in any way including, but not limited to, textual or binary changes, and distribute it to any other person or organization.
2. Continue to distribute this program if you receive written notice from me to stop.

2. Acknowledgements

Great thanks first of all, my wife who put up with her husband coming to bed late, and disappearing at unusual times, or suddenly, with an odd expression, scribbling something down on whatever crumpled piece of paper he could find.

Also thanks to Ed and especially JoAnne, who put up with the original user interface, and with "Beta releases" every 2 or three days. I appreciate greatly JoAnne's ability to find ways to easily beat the computer as quickly as I could find ways to thwart them.

Finally, thanks to my employer who allowed me to take home the SDK which got the project started (it was perfectly legal, by the way -- I deleted it from my

hard drive at work, since I wasn't using it at that time).

3. Introduction

Several years ago, there was a board game that was similar to CONFOUND. I loved playing it, but, since it wasn't your run-of-the-mill strategy game, in order to play it with someone, I had to first teach that someone the game. Of course, once that happened, I could have them for lunch since I knew the game very well, and they were just novices.

Well, time passed and, like a lot of other things I've enjoyed, the game went off the market and the company that did it went out of business (as far as I can tell-- I've looked, and can't find them anywhere).

When I started getting into Windows programming, I looked around for something to program, and thought of this game. This way, you start off with a knowledgeable opponent -- the computer itself.

This game will run on anything that runs Windows 3.0 or 3.1. An 80386 or above is recommended for good response time. A color monitor is also recommended.

To install the game, copy CONFOUND.EXE, CONFOUND.WRI and CNFNDREG.WRI onto a convenient directory on your hard drive. If you can't think of any better one, your WINDOWS directory will do fine.

To start the game, either go into File Manager and double-click on CONFOUND.EXE in the directory you've copied it to, or install it into the Program Manager using the "NEW" command, and then double-click on its icon.

4. Playing the game.

I've spent a lot of time on the Help facility in the game itself. Rather than spend a lot of time again repeating all that, take a minute to fire up the game and read the Help messages. I'll wait.

Tum de dum de dum de dum...

Done? Good.

Now let me see if I can expand on the Help messages. Both you and the computer start off with 2 "pegs" and 2 home squares. You start off on your home squares, ditto for the computer.

Each move you both have a choice -- build or move. To win the game, one of you needs to get one of his pegs onto one of his opponents home squares. This means, you must make it as easy as possible for you to get there, while making it as difficult as possible for your opponent.

I sweated over the user interface, since the actions involved are reasonably complex. The principles to remember are:

1. The cursor will change shape to help you know what to do next.
2. To do things on the board, you press and *hold* the mouse button. To pick up things off the board, you click and *release* the mouse button.

This means that, to move the peg, you press the mouse button on the current location of the peg, and don't let go until you reach the square to which you wish

to move.

To place a wall, you click on one of your walls in your "pile," and then release the mouse button. You then go over to the board, and press and hold the mouse button over the first square into which you wish to place a wall. You move the cursor up, down, left or right, still holding the mouse button, until the wall is where you want it, then release the button. A wall is always built on the bottom or right edge of the square you select. Walls cannot intersect existing walls, or block paths to home squares.

5. Winning the game

There are three levels of difficulty in the game. The first two use minimal long-term strategy, and the "Hard" level uses every trick in the book. I wish there were an easier transition, but I couldn't think of any.

There is a bit of pure randomness to the computer's strategy (by intent), so that it won't play exactly the same game each time you play.

There may be other strategies than these, but the main kinds of walls the computer deals with are:

1. Offensive walls. These are used simply to block it's opponent's path while leaving it's own unscathed.
2. Back walls. These are used to narrow the possible paths to only the best one, which then cannot be blocked because it's the only way to get to a home square.
3. Defensive walls. These are used to block a particularly damaging offensive wall that its opponent could place.
4. Tube walls. A tube is an unblockable path between a peg and a winning home square. Right now, the computer tries to prevent you from constructing a tube, but it isn't smart enough yet to build tubes of its own. See the next section.

Don't worry about using all of these to start with; take it easy, learn the game, and have fun.

I find the game is fun whether I win or I lose. It's hard to judge how hard the game really is to play; remember, I've eaten, slept, and breathed this game for the duration of the development effort (about a year of time). Let me know how you find it -- whether it's too easy, too hard, or just right. One indication I have is that, even if I win, I need to stay on my toes -- one mistake, and the computer chews me up and spits me out.

6. Future directions

I believe it was Carnegie who said, "there comes a time to shoot the engineer and go into production." I agree wholeheartedly; I could diddle with the strategy of this thing until I retire, and still not be 100% satisfied. Then, when I did finally release it, find out only 3 people ever like it enough to register it.

In order to prevent this, I decided to release it now. It behaves itself, doesn't make any psychotic moves that I can detect, and plays a decent game. If enough

people register it, where I would like to see it go (no commitment on this) is (not necessarily in the order they'll get implemented, either):

1. Right now it does a decent job of preventing "single-width tubes." It completely misses "double-width" tubes. This is mucho worko indeedio.
2. It needs to build its own tubes. This gets real heavy real fast.
3. Like chess, the game is won or lost in the opening. This means either an "opening book" or a neural net of some sort. I actually have the neural net code designed, but what terrifies me is training it and debugging it. I'll spend more time doing that than in implementing it. My leaning right now is to come up with a way to have the neural net play the existing heuristic method so it could teach itself.
4. The computer would play better with multiple-move lookahead. This means that it either needs to be algorithmically faster (I have some ideas in this regard), or I just wait for the next generation of microprocessors to hit the street in a big way. It's also a lot of work.
5. I've toyed with the idea of making it possible for two humanoids to play each other, rather than against the computer. My thoughts on this is to add facility to play on a network using DDE or something similar. This would be tough, since I don't have a network right now to debug it on.

7. About the author

I often wonder with Shareware whether I'm dealing with a roomful of programmers or one guy in his basement in his spare time. As far as CONFOUND is concerned, wonder no longer -- as I write this, I'm sitting alone in a basement, listening to a washing machine running in the background.

My "day job" is an Engineering and Design Manager at a major metropolitan newspaper. I have worked there for seven years now.

We have undergone frequent reorganizations in the last 18 months. For several of those, it looked like my "hands-on" days were over. One of the reasons I started work on CONFOUND was because I wanted to keep my "hands in" just-in-case. Out of work technical managers are everywhere these days, but GUI programmers are still much in demand. In addition, I needed to know if I still had the "right stuff" as far as development was concerned.

Lately, however, due to the magic of downsizing, I find myself running a department with 2 engineers working for me, and more than enough work for 3. This means I'm now very hands-on, but doing most of my work with a different, but similar GUI, namely the Macintosh. My work with Windows has made it easier to get over the steep Mac learning curve, since I just needed to understand the differences from Windows, rather than starting from zip.

I also have been watching the plight of my fellow (and seemingly expendable) engineers with a great deal of concern. It struck me that the best thing I could do is to get off my duff and move into a direction that might result some day in the creation of jobs.

Personally, I have a wife and a toddler son. We live in a little single-family

dwelling just outside the city.

My hope for CONFOUND is that it is moderately successful. Certainly, some of the proceeds from it will be plowed back into my computer, to add more disk storage to better accomodate the modern development behemoths. I have 2 20 Meg drives, one of which is totally dedicated to my C Compiler, and the other about 50% dedicated to Windows itself. Sigh.

The machine that this was developed on is a "nameless Taiwan clone" PC, which started as an XT six years ago, and now is on its third motherboard, a 16 MHz 386SX with 3 Megs of memory. My goal was to make CONFOUND run acceptably on this, which is about the minimum for a "serious" Windows machine. This means that it should be blazingly fast for you lucky ones with the 486DX-2/66 machines.

If it makes more money than that, I've toyed with the idea of a laptop so I could develop things upstairs where the real people live, and/or a small Mac so I could write some things for that market. You might even see a Mac version of CONFOUND some day.

Over the long term, the proceeds will also find their way into my son's college savings and/or moving into larger dwellings to accomodate aging parents on both sides.

Of course, if this turns out to be the software "pet rock" of the 1990s, we're off the ground as far as a "real" company is concerned.

I hope you do enjoy this game. It's been both fun and a real pain to write. I'm glad you at least tried it out.

I've got a notebook full of other shareware ideas. Keep your eyes open.