Score4

Ian D. Romanick

COLLABORATORS			
	TITLE : Score4		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Ian D. Romanick	October 23, 2022	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

Score4

1.1 Score4

```
                                  Score4 by The Dancing Fool
                        (C) 1995 Epsilon Software


                          How To Start
                       - Running and installing the program

                          How To Play
                       - Introduction to playing the game

                           How To Win
                       - Tips for winning

                             History
                       - How and why the game was made

                      License  Agreement
                       - Read this before you use the game
```

1.2 Starting a Game

Before you can actually play the game, you need to select which binary to use. In the current release there are two included. One is for 68000 users and the other is for 68020+ users. If your Amiga has only a 68000 or a 68010, you will need to use the "Score4.68000" binary. If you have a 68020 or better, you should use the "Score4.68020" binary.

The easiest way to start the game of Score4 is to just double-click on the icon. Once this has been done, a requester with four gadgets will be presented. Each of these gadgets corresponts to one of the following play modes:

```
H vs H  - Human versus human
H vs C  - Human (player one) versus the computer (player two)
C vs H  - The computer (player one) versus human (player two)
```

C vs C - The computer versus the computer

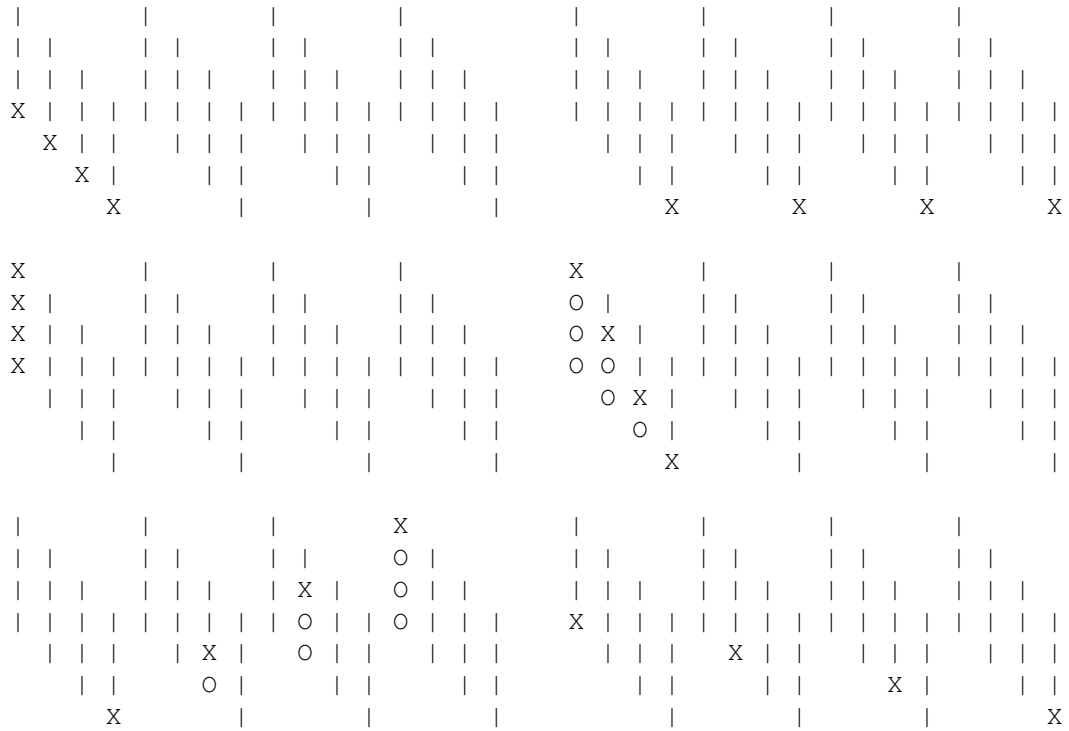
Score4 can also be run from the CLI.

1.3 How to Play Score4

Score4 is a simple strategy game that plays much like the classic game "Connect Four." In this game, however, there is a twist. Rather than playing on a flat, two-dimensional board, Score4 is played on a three-dimensional board.

Players take turns selecting one of the 16 pegs to place a piece on. Once a piece has been placed it drops to the lowest of the four potential positions on the peg. Once a peg has been filled with four pieces, it can have no more pieces added to it. A move is selected by clicking on one of the 16 gadgets located below the game board in the Score4 window. Once a peg is full, its gadget will be ghosted to show that it is no longer available.

The game is over as soon as one of the players manges to get four of their pieces in a row. The six possible ways that this can happen are depicted below.



1.4 Tips and Tricks for Winning

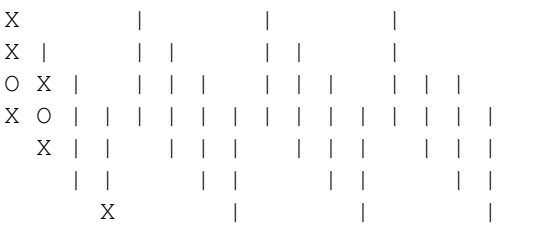
The strategy involved in the game of Score4 is surprisingly complex. There are a number of subtle issues surrounding the winning techniques of this game that I will attempt to cover in this section. Please keep in

mind, however, that I am by no means an expert at this game. I have actually only been playing it for about three weeks.

There are two fundemental techniques that frequently lead to victory that I call "traps" and "multiples." These techniques are actually not unlike "discovered check" and "forks" in chess. If you are familiar with these techniques, you will have a strong advantage in Score4.

Traps

A trap is perhaps the most dangerous construct in the whole game. It is also, however, one of the most difficult to set up. The basic priciple is that in order for a player to block a four in a row on the current turn, he will set up another for the next turn. For example, consider the trap that the X player has set here:

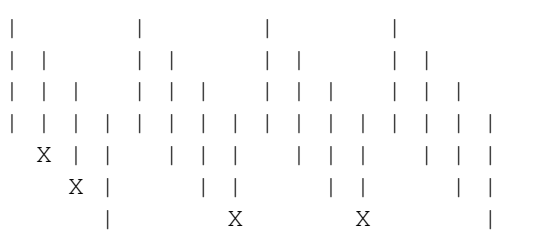


In order for the O player to avoid defeat on the current turn, he must block at position (1,0). However, by doing so he sets the X player up for a win on his next turn.

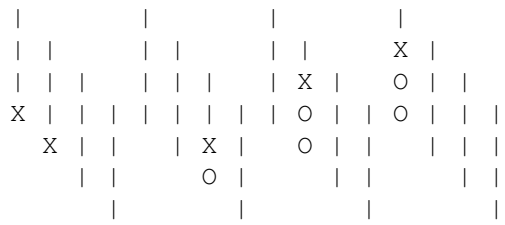
Fortunatly the computer AI does not often set traps. When it does so, it is purly coincidental. By the same token, the AI can usually prevent a trap from being set. The quickest that I have ever been able to set a trap for the computer ended the game on round 16.

Multiples

The multiple, or multi-way, is another sure fire road to victory that is very difficult to set up. The priciple here is that by making one move a player creates two unblocked three in a rows. This creates a situation where the other player would need to make two block with one move. Clearly a difficult situation. Consider the following game board:



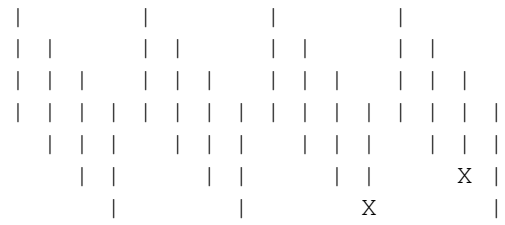
By moving to position (0,0), the X player will assure vicory on the next round, assuming that the O player doesn't obtain victory first. The example situation is very unrealistic. I would have to assume that a player who would get into such a situation must be drunk. There are, however, very realistic situations where the multiple comes into play. For example, the following board shows the X player not only blocking the foolish O player, but snatching victory right out of his hands:



Other Hints

There are a couple of other techniques that I have noticed often lead to victory. The most obvious is the occupation of the corners. I have noticed while watching computer versus computer that the player who occupies three or four of the lowest level corners wins nearly 90% of the games. The reason for this appears to simply be that there are more ways to win with the use of the corners.

Another issue is the selection of a good set of opening moves. I have found that it is very useful early on to use a set of moves that give a good spread of the game board but that don't build any two or three in a rows. One open that I find works very well agains both humans and the computer AI is the following:



At this point the selection of the next move depends totally on what the other player has done with his first two moves. However, the next move is typically either (3,0), (0,3), or (1,2). Often if I use (3,0) as my third move, my fourth will be to either (0,0) or (3,3). The general tactic is to not build anything until a large number of things can be build with one move. The result is that the other player needs to use a lot of moves to block a large number of potential victories.

This section is quite probably both the most complete and the most incomplete in all of the Score4 documentation. I think that it will provide a good start for developing winning techinques for this game. As I play this game more and hear from other who play, I am sure that this section will continue to grow.

1.5 History of Score4

The concept-to-completion for Score4 is kind of interesting. It all started on August 3rd, 1995, when Steve Chapel (schapel@zonker.cs.ucsb.edu) posted a challenge to comp.programming.contests detailing a simple game played on a 4x4 grid of posts. The basic idea was to implement a version of this game, which he called "Score Four," that was unbeatable. I thought that the idea was sort of interesting, so I filed the article away.

About a week after that, a friend of mine came to me asking to suggest

a simple project for him to write. I immediately thought of Score4. I suggested that he write a simple two player version of the game, then possibly implement AI for it. At that time I decided that I should also write a version of the game so that I would be able to offer real advise, should he need it.

I started writing the game that night, and in just over two hours I had a fully functional ASCII graphics version of the two player game. I spent about thirty minutes playing the game against myself to make sure that all of the routines were working correctly. In that short time I came to a startling realization: this game is fun!

With this in mind, I decided that I had to write AI so that I would have someone to play the game against! The AI took about eight hours to write, and seems to work pretty well. Once I had spent this much time on it, I decided that I might just as well make a GUI and some documentation for it and turn this simple project into some sort of a product.

After all that, the game that you have before you exists. It's not perfect, but it is here! The current version of the AI uses a simple minimax algorith and a very simple heuristic. The next version should have a better heuristic and use Alpha-Beta pruning. The GUI uses only gadtools.library and intuition.library, and is totally font sensitive. The program was written using GCC 2.6.

As you may have already noticed, there are two versions of this program. There is one for 68000's and one for 68020's. The 68000 version is included only for completeness. Since it takes about 9 seconds to generate a move on my A3000 25, I do not even want to think about how long it would take on a stock A500. I would, however, be interested in hear about how long it take the AI to generate moves on various configurations.

Enjoy!

1.6 License Agreement

This program is a shareware product. I you enjoy playing Score4, please send either $5 or a registered version of some piece of software that you have developed to:

Ian Romanick
3745 SW 108th Ave, #6A
Beaverton, OR 97005-1849
U.S.A.

In any case, please send me a piece of e-mail and tell me what you think! My address is:

idr@cs.pdx.edu

The latest version of Score4 should be available on Aminet and on my web page at:

http://www.cs.pdx.edu/~idr

This software may be distributed freely so long as all of the following listed files, and those files only, are included in the archive, and no fee beyond a nominal media fee is charged.

```
Score4
Score4.info
Score4.guide
Score4.guide.info
readme.txt
```

EPSILON SOFTWARE DISCLAIMS ALL OTHER WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OF NON-INFRINGEMENT WITH RESPECT TO THE SOFTWARE AND THE ACCOMPANYING WRITTEN MATERIALS.

IN NO EVENT WILL EPSILON SOFTWARE BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY LOSS OF PROFITS, LOST SAVINGS, OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF YOUR USR OR INABILITY TO USE THE PROGRAM.

THIS SOFTWARE IS DISTRIBUTED AS-IS.