Go-Moku 2.1

Shareware Intelligent Game

Help Contents:

History of Go-Moku

See also:
Acknowledgments

1. Go-Moku

The ancient Japanese game of *go-moku* (five stones), still popular in the Orient, is played on the intersections of a *go* board (this is equivalent to playing on the cells of a 19x19 square). Players take turns placing counters from an unlimited supply until one player wins by getting five in a line, orthogonally or diagonally. Experts are of the opinion that the first player can force a win, but as far as I know, no proof of this has ever been published.

The game became popular in England in the 1880s under the name of "go-bang". It was sometimes played on an ordinary draughts board, each player using 12 or 15 draughtsmen. Moves were permitted in any direction if no one had won by the time all the draughtsmen were placed.

2. About this implementation

The original algorithm, which was developed in January 1994, assumed deep search of the best move, say 3 - 5 moves ahead. However, when the first version of the game was brought to life by Konstantin Gredeskoul, it was unexpectedly found that the algorithm based only on estimation of the current position achieves amazingly high level of playing.

Thus, the algorithm implemented in this version is "*today-oriented*", i.e. program only takes into account what is good at this moment - some sort of *Greedy Algorithm.*

If any response is received from users, it is possible to extend this game so that it looks through different combinations in the future. I believe that this would *not* significantly improve playing performance. However, my colleagues are of a different opinion. Well, it's only a matter of time to decide.

Game Rules

See also:
Game Hints and Tricks

This program plays a very old Japanese game called Go-Moku (also known as 5 in a line). The original board has 19x19 squares (27x27 in this version). Get 5 stones in a row (diagonally, horizontally or vertically) and you win!

This game is for two people; one plays crosses, second plays noughts. Game always starts with a cross. Game is over if one of the players puts five in a row.

Well, even if there is an always-winning algorithm for this game I am not aware of it. The program uses best-move estimation and chooses at random one move among several moves of equal cost. So, you always have a possibility to win, even while playing on the hardest levels.

Program Environment

Menu commands:

1. **Game**
2. **Level**
3. **Options**

Product Information

See also:
Shareware Concept

Go-Moku 2.1 © 1993-1995 Konstantin Gredeskoul

Go-Moku is marketed under the **Shareware concept.** You are free to use this program for a trial period of two weeks. If you continue to use this program after the trial period you are required to register it. The program is registered by mailing in the registration form with the appropriate fees. You may print registration form via <print registration form> command in Help Menu.

For Australian users, fee is **$20(AU)**.
For users in other countries, fee is **$20(US).**

Personal cheque, money order or Bank cheque are the only acceptable payment methods.

Only Australian and American dollars are accepted.

If you want your copy to be delivered by Air Mail, then please add $5. Other available delivery methods include FTP, World Wide Web and E-mail (e.g. sending uuencoded file).

Please, send your check or money order to

Konstantin Gredeskoul
4/7 Exhibition Street,
Bentleigh, VIC 3204,
AUSTRALIA

If paying by check, please make them payable to Konstantin Gredeskoul and protect both you and me by writing "*Not Negotiable*" across your check, and don't forget to indicate what currency you use.

If you register, you will be supplied with the last updated version of Go-Moku registered *on your name*. You will also be eligible for free upgrades in the future. All your questions will be answered in details. You will be supplied with a list of other developments of Konstantin Gredeskoul, offered to you at 25% discount as a licensed user. And the last reason to register is to support one more Shareware programmer.

Please, send any comments, questions or encouragement to either postal address above, or to my Internet locations:

E-mail 1: <kig@brain.physics.swin.oz.au>
E-mail 2: <kig@fermat.maths.monash.edu.au>
WWW: http://www.maths.monash.edu.au/~kig

This information is valid at June 22, 1995, and will remain valid for, say, the next couple of years.

Shareware Concept

If you are familiar with the idea behind Shareware, then you know that Shareware is the ultimate in money back guarantees.

Most money back guarantees work like this: you pay for the product and then have some period of time to try it out and see whether or not you like it. If you don't like it or find that it doesn't do what you need, you return it (undamaged) and at some point - which might take months - you get your money back. Some software companies won't even let you try their product! In order to qualify for a refund, the diskette envelope must have an unbroken seal. With these "licensing" agreements, you only qualify for your money back if you haven't tried the product. How absurd!

Shareware is very different. With Shareware you get to try it for a limited time without spending a penny. If you decide not to continue using it, you throw it away and forget all about it. No paperwork, phone calls, or correspondence to waste your valuable time.

Software authors who use the Shareware method of distribution feel that Shareware is the best way to try a product. You are able to try it on your own systems, in your own special environment, with no sales people looking over your shoulder. Have you ever purchased a car and realised that is you could have test driven it for 2 weeks your purchase decision might have been different? With Shareware these problems can be avoided - you do have a 2 week test-drive!

After trying a Shareware product and deciding to continue to use it, then - and only then - do you pay for it. Not only that, but Shareware is traditionally much less expensive simply because you are paying for the software, not the advertising and marketing that comprises that majority of the cost of most software (a one page ad in PC Magazine, one time, can cost upwards of $20,000). If the try-before-you-buy concept sounds like an ideal way to make your purchase decisions, you're right!

Some companies burden their products with annoying copy protection schemes because they don't trust their users. Shareware developers not only don't use copy protection, they freely distribute their products because they do trust their users. Someone once said, that you should never trust software that doesn't trust you. This makes a lot of sense - no wonder Shareware is becoming so popular among users and developers.

Shareware is a distribution method, not a type of software. Shareware is produced by accomplished programmers, just like retail software. There is good and bad Shareware, just as there is good and bad retail software. The primary difference between Shareware and retail software is that with Shareware you know if it's good or bad before you pay for it. Registration of Shareware products, in addition to being required, is also an incentive for programmers to continue to produce quality software for the Shareware market.

There is another significant advantage to Shareware - it allows small companies like Bright Mind Software to make software available without the hundreds of thousands of dollars in expenses that is takes to launch a traditional retail software product. There are many programs on the market today which would never have become available without the Shareware marketing method. Please, show your support for Shareware by registering those programs you actually use and by passing them to others.

Thank you for your support!

Game hints and tricks

This section is aimed to explain some elementary combinations arising during the game and possible strategies in response to these combinations.

Couple of *definitions* before we begin:

Open triple/quadruple is a combination of three/four crosses (noughts) in a row, with a possible hole inside, such that its both sides are open (not occupied)
Closed triple/quadruple - at least one side of triple/quadruple is closed by an opponent.
to close an opponents combination (eg triple, quadruple) is to occupy one of the two empty adjacent to opponents triple/quadruple positions.

Let's suppose that we are playing **O**, the other player is playing **X**. If your opponent puts three **X**s in a row open from both sides, then it is equivalent to a check in Chess. If we have no possibility to put five or open quadruple, we must close his three crosses from either side, for otherwise the next his move will create four crosses in a row open from both sides. Open quadruple is a mate, because you can close it from only one side at a time. All of the above results in a very simple rule: *if you have no extremely good combination to create - close the opponents open triple.*

Example: (Diagonal triple of **O**s open from both sides)

```
X OO
OXOX
 O X
```

Another useful combination to know is an open triple intersected with a closed from one side quadruple, both of the same player (e.g. both in **X**s). This almost always results in a mate. Your opponent has to close an open side of the quadruple; you can then build an open quadruple from the remaining open triple. This is a mate, as we know from the previous paragraph.

Example: (diagonal **X**-triple intersecting open from one side quadruple of **X**s)

```
X O
OXOO
 OXXXX
```

Similar position arises when you have two open triples. But these can sometimes be closed by opponent's four which you in turn will have to close, thus, loosing activity. A stronger combination is two *half-open* (i.e. closed from one side) intersecting quadruples. Closing one still leaves you with an opportunity to put five from the remaining four.

The main feature of this game is that there are infinitely many combinations (well, for 27x27 board there are exactly 729! / (364!x365!) combinations - enough for several human lives at least...) so you may enjoy it again and again!

Acknowledgments

Author is very grateful to

- **Igor Zilberman**
- **Valery Giner**
- **Eugene Miloslavsky**
- **Maria Gredeskoul**

for their participation in discussions and development part of Go-Moku.

This implementation is by

- **Konstantin Gredeskoul**

June, 1995

Menu Commands: Levels

Computer can play on three levels: 1 - as a beginner, 2 - as an intermediate player and 3 - advanced.

The primary difference between them is that at level 1 computer plays as a very risky and self-confident player, who does not pay too much attention to your combinations. It only looks for it's own good moves. Advanced level is much more difficult, because computer becomes very thoughtful and makes moves with care. Intermediate level is somewhere in between these two.

None of the levels are unbeatable, and for all of them you will need some thinking in order to win.

Menu Commands: Game

New / Stop
New command clears the board, starts a new game. If the game is started, this command reads as **Stop**, which stops the game and reactivates some options which are disabled while game is being played.

Move Back
Very useful command. It works differently in "Human vs Computer" mode than in "Human vs Human" mode:

> If you play with computer, then *Move Back* command gives you an opportunity to re-think your last (and only last) move. So, if you did a wrong move and computer responded with another move, pressing <Move Back> button (or menu command) will take both yours and computer's moves back.
>
> If you play with your friend, then <Move Back> command takes back one move at a time and goes as far back you wish.

Exit Go-Moku
Quit the program

Menu Commands: Options

Note: Some of the commands listed below are of the Boolean nature. They are either ON or OFF. ON state is indicated by a "tick" to the left of the command.

Human vs Computer
If this is set to ON, you play against computer.

Human vs Human
Alternatively, you can play with someone else, using the program as a board.

Computer plays as - X/O
Obviously, this command is aimed to assign X or O to your computerised partner.

Enable keyboard usage
If this is ON, then you can use keyboard arrows to move pointer. Press <Enter> or <Space-bar> to make a move. Very useful feature when two people are playing. Then, one can play using a mouse and another player can play using keyboard.

Centre Board
This command moves game window to the centre of the screen.

Save Settings
This command saves current settings including:

- Current date and time
- Position of the window on the screen
- States of the options above (e.g. level, who plays, enable keyboard or not, etc.)

Warranty

The author hereby disclaims all warranties relating to this software, whether expressed or implied, including without limitation any implied warranties of merchantability or fitness for a particular purpose. The author will not be liable for any special, incidental, consequential, indirect or similar damages due to loss of data or any other reason, even if advised of the possibility of such damages. In no event shall the author's liability for any damages ever exceed the price paid for the license to use the software, regardless of the form of the claim. The person using the software bears all risk as to the quality and performance of the software. Use of this package constitutes agreement on the part of the user to this disclaimer.