COMP3211 Project
Playing Gomoku: To Win and To Play For a Long Time

Due Time: 23:59 on Sunday, 22 April, 2018

1 Introduction

Traditionally, Gomoku, also called Five in a Row, Connect 5 or Gobang, is a board game played by two persons on a 15x15 Go board[1]. One plays with black stones and the other with white stones. Initially the board is empty, and the black plays first, then the players alternate in placing a stone of their colour on an empty intersection. The winner is the first player to get an unbroken row of five stones either horizontally, vertically, or diagonally. The game ends when one of the players has won the game or when the board is full and no one has won, i.e. it is a tie. Since 2000, there has been annual tournaments of Gomoku by computer programs. For details, see http://gomocup.org/.

In general, Gomoku can be played on any NxN board. In this project, you are going to write computer programs to play on 8x8, 15x15 and 19x19 boards. However, instead of winning as the only goal, we want you to try to play at the level of your opponent. It is obviously difficult to define what it means to play at the level of the opponent. In this project, we will just use the length of the game as the indication of how closely the two players match against each other: the longer the better. More precisely, we will use the following scoring function:

$$s = s_0 + \frac{L}{N \times N}, \tag{1}$$

where s_0 is 1 if you won the game, -1 if you lost, and 0 if the game was a tie; and L is the total number of stones (both black and white) on the board at the end of the game, and N is the dimension of the board.

2 Platform

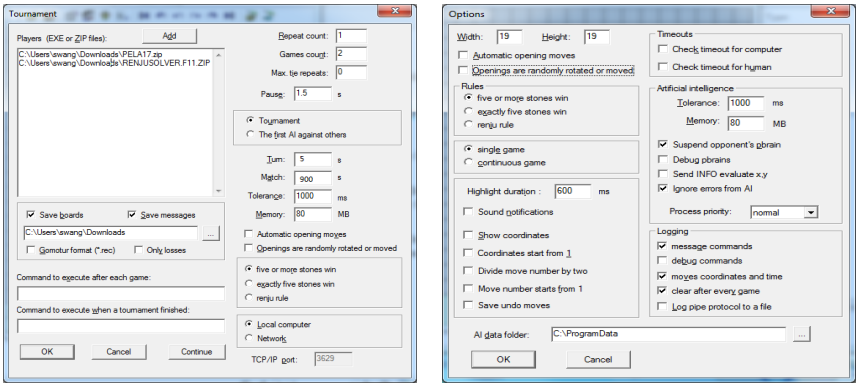
The project will use the Piskvork manager which can be downloaded here:

http://gomocup.org/download-gomocup-manager/

[1] http://gomokuworld.com/gomoku/1

This manager runs on Windows platform. It's light weighted and can even be run on our university Virtual Barn (http://itsc.ust.hk/services/academic-teaching-support/facilities/virtual-barn).

The following are some screen shots of how to configure the manager to run a tournament:



(a) Tournament (b) Options

Figure 1: Tournament Setting

You can experiment with this manager using computer game players available at

http://gomocup.org/download-for-developers/#templates

We will also do a tutorial on installing and running the Piskvork manager.

3 Tournament and marking scheme

The marking scheme of the project is as follows:

1. **20%** for report and/or presentation. You will need to hand in a report that describes clearly your algorithm behind your program. There is no fixed format for your report. The main thing that we look for in your report is how you design your program and how well you describe your algorithm. We will select about 15 students, each to do a 5-minute presentation of their project in class, tentatively scheduled on the last class of the semester. If you are selected, then you'll get at least 90% of this portion of the mark if you do your presentation without any major problem. If you are selected but fail to do the presentation without prior approval from us, you'll get 0 for this portion of the mark.

2. **80%** for how well your program performs in the tournament. The tournament will be run on three types of boards: 8x8, 15x15 and 19x19. For

each type of the board, we will run your program against all other students' programs plus a few programs written by us. Your program will play twice against each of the other programs: first as black and then as white. For each of board types, your program will receive a score which is the sum of your program's scores according to (1) in all matches. Thus your program will receive three scores in the end, and the final score of your program will be a weighted sum of these three scores. We have yet to decide on the exact weight of each board type. The final score of your program will then be used to compute how much you get for this portion of the mark. We will let you know the weights and the formula that we use to compute your grade in the end.

4 Submission

You must submit your project as a zip archive file on canvas. The file should be named as 'comp3211-YourStudentID.zip', for example, 'comp3211-12345678.zip'. Your archive should contain at least the following files,

- Source code of your program. We will check our code to satisfy ourselves that it is original. If you have done an originality analysis of your code on your own, you can include a copy of the report. Notice that this does not mean that you cannot use open source codes. It's just that if you do use them, you have to acknowledge it in your report and convince us that there is something substantially new in your code.
- Your program executable (.exe) file(s). If you have different programs for the three different types of boards, name them as follows: 'pbrain-YourStudentID-8x8.exe', 'pbrain-YourStudentID-15x15.exe', and 'pbrain-YourStudentID-19x19.exe'. If you use the same program for all of them, name it as 'pbrain-YourStudentID.exe'. Please test your program on the Piskvork manager to make sure that it runs without any problem. You can even test it by matching two copies of your program against each other.
- Project report in pdf format. As mentioned above, please describe the algorithm behind your program clearly. If you made use of any open source code, please say so in your report. It's also a good practice to include this information as comments in your source code.
- Project presentation slides in pdf format, maximum 5 pages. As mentioned above, you may be asked to give a 5-minute presentation in class about your program.

5 Reference

Finally, below are two pointers to resources that may help you with the project:

1. A summary of good resources on gomoku/renju, especially for designing programs to compete in the Gomoku Tournament, written by Kai Sun:

http://www.aiexp.info/gomoku-renju-resources-an-overview.html

2. Gomoku AI related articles provided by Gomocup: http://gomocup.org/links/