SapioGo
Project Requirements

Ryan Flannery
`squid@fuse.net`

Kevin Upchurch
`upchurks@email.uc.edu`

Advisor: John Schlipf
`john.schlipf@uc.edu`

June 8, 2004

Contents

1 Introduction

The game of Go is an ancient game that is considered to be far more mentally challenging than chess, yet has only a few simple rules. With such a simple set of rules, many would think that modeling the game on a computer would be straightforward, and that it would be easy to construct a computer capable of beating even the best human Go players. To date, however, this has proven to be exceedingly difficult and most attempts have produced Go games that even a novice can easily defeat.

The goal of "SapioGo" is to engage others in a game of Go and, over time, learn to play a *better* game of Go.

2 The Game of Go

2.1 History and Evolution of the Game of Go

The game of Go is believed to have originated between 4,000 and 5,000 years ago in the area that is modern-day China and is considered to be the oldest game still played today. Mystery surrounds the creation of Go, but some believe that Go may have been a precursor to the abacus. One legend tells of an emperor who created the game in an attempt to make his dull-witted son more intelligent.

By approximately 600 B.C., Go had become one of the "Four Accomplishments," which included brush painting, poetry and music, that all Chinese gentleman were to master, thus playing a major role in Chinese culture.

Given China's close proximity to countries such as Japan and Korea, it's not surprising that Go became equally prevalent in those countries. In the early 1600's, four schools of Go were created in Japan, where students learned to master the game in a formalized setting. These schools still exist today and are a major part of Japanese culture. Japanese Go Masters are considered celebrities in their country.

Today Go is most popular in Korea where it is believed that between five and ten percent of the population plays regularly. Some of the worlds best Go players reside in Korea.

Go was brought to the United States in the mid-1800's when multitudes of Chinese workers were immigrating to the U.S.

With so many different cultures playing the game of Go, it's no wonder many different methods of playing evolved. Each culture has its own rule system, varying in the way points are tallied at the end of the game. Some even have their own timing systems which force players to make moves in a required amount of time.

2.2 The Rules of Go

Go is a game played on a large, flat wooden board where two opposing players place white and black stones called "ishi" on the board. The board is square with 19 horizontal lines intersecting 19 vertical lines. Although Go games may be played on any size board, 9x9, 13x13, and 19x19 boards are the most common. The two players alternate placing stones on the intersection of two lines, rather than in the squares formed by them.

When the stones are placed on the board, players have the option of trying to *capture* the stones of their opponent. Capturing the stones of one's opponent is done by entirely surrounding the stone of one's opponent. Once this is done, the opponents' stone is removed from the board. If numerous stones of the opponent are adjacent to each other, then all of the stones must be surrounded and captured at once.

The goal of a game of Go is to encircle territory on the board. This is done by surrounding large areas of the board with stones and preventing the opposing player from doing the same. Also, players want to minimize the number of stones of their color that are captured while trying to capture as many stones of the opponent as possible.

The Black player always moves first in a game of Go. This provides an unfair advantage to the Black player, and to help compensate, the White player always receives an additional $5\frac{1}{2}$ points at the end of the game. This rule is called *Komi*. The amount of Komi given to the White player is the same regardless of the board size. Also, the additional $\frac{1}{2}$ point that is awarded to White makes it impossible to have two players tie in a game of Go.

Each player has the option of passing when it is his/her turn, opting not to place any stones on the board. When both players pass consecutively, the game is considered over. This usually happens when there is no territory left on the board for one of the players to capture, or when it is obvious to one of the players that their any continuing effort would be futile.

Once the game is over, the score for each player is tallied and the player with the higher score wins. Methods for determining the score vary with the rule system being used, but they all roughly award points as follows: every intersection on the board that is completely encircled by stones of one color are awarded to that player. Every stone on the board that is responsible for encircling some territory is also added to that players score. *Note that stones on the board that do not take part in encircling some area are called Dead Stones and do not affect the score.* Stones that have been captured by a player are added to that players score.

2.3 Terminology

Below is a brief introduction to some common terminology used in the game of Go, as well as throughout the documentation for this project.

Atari An state when your opponent can capture one or more *ishi* in the next move.

Dame A point on the board that is useless for both players.

Fuseki The opening moves in a game before the real strategy begins.

Gote A move causing the current player to lose *initiative*. This is the opposite of *sente*.

Initiative A state where the current player can force their opponent to move in certain locations.

Ishi Used to refer to the stones in a game of Go.

Ko A situation where a single board position may be captured and recaptured infinitely by both players. *The rule of Ko* eliminates such endless cycles by not allowing a previous board state to be repeated.

Komi A number of points (usually $5\frac{1}{2}$) that is added to White's final score to compensate for Black having moved first. Some rule-sets may choose to use a different amount of points or none at all. Note that the $\frac{1}{2}$ point prevents a Go game from ever ending in a draw.

Kyu A standard ranking used to evaluate weaker players. The ranking ranges from 1 to 25, with 1 being the strongest and 25 being the weakest.

Sente A state that maintains *initiative* for the current player by forcing their opponent to move in a certain location since any alternative move would result in the loss of critical *territory*.

Territory Area on the Go board where both players battle for control.

2.4 Computer Go

The game of Go has a long history in the field of computer science. Its unique nature has made it very difficult to model with any algorithm and no existing approach has created a Go client capable of consistently beating even a novice.

The field of study focused on creating an intelligent Go game is known as *Computer Go,* and includes work in areas such as algorithm design & analysis, artificial intelligence, and data structures.

Computer Go is so well established that standards for file formats and communication protocols (for playing Go over a network) have already been established. There are even numerous open-source software packages available that implement these standards, allowing the development of intelligent Go systems to focus more on the design and implementation of the intelligence.

3 Artificial Intelligence

3.1 Basics

By default, SapioGo will take no more than 10 minutes to determine its next move during a game. The user may, however, specify a longer time to allow SapioGo to determine its next move and thus allowing it to decide on a better move. As the game progresses, SapioGo will learn to make better moves and hopefully provide more of a challenge to its opponent. SapioGo will also be able to store what strategy it has learned from one game, and apply it to the future games.

3.2 Algorithms & Approaches

SapioGo will use a mixture of intelligent algorithms. What algorithms and approaches it takes has still yet to be decided. SapioGo will most likely use a genetic algorithm, implemented using neural nets, combined with a heuristic look-ahead algorithm. What algorithm is employed for each move will be based on a hierarchy of requirements. For example, if there are locations on the board where a single move by the opponent may result in the loss of considerable board territory or stones, then SapioGo will employ its heuristic look-ahead algorithm to decide on its next move. Otherwise, SapioGo will employ its genetic algorithm to determine its next move.

4 Project Requirements

4.1 Game Details

SapioGo will support playing a basic game of Go with traditional Chinese settings. SapioGo will allow either itself or its opponent to make the opening move, and will also support handicaps for either its opponent or itself.

4.1.1 Board Sizes

SapioGo will support the standard Go board sizes of 9x9, 13x13, and 19x19, and will even allow the user to specify an arbitrary board size. During development, two species of SapioGo will be evolved: the first will use a 5x5 board and the second will use a 9x9 board. Although the 9x9 board is only slightly larger than the 5x5, the neural network required for the 9x9 species will be over 6 times larger than the one required for the 5x5 board. As such, the evolution of the 5x5 species will be much faster and hopefully provide insight into the evolution of the 9x9 board.

4.1.2 Rule Set, Time System, and Handicaps

SapioGo will only be able to play using the standard Chinese rules of Go and will not use any timing system. It will support a standard Komi of $5\frac{1}{2}$. Also, SapioGo will not support handicaps for either player in a game.

4.1.3 Opponents

SapioGo will be able to play against human opponents or even other automated Go clients. It could even be set to play against itself. SapioGo will also support playing opponents over a network, allowing users to play SapioGo from remote locations.

4.1.4 Benchmarking

During testing, SapioGo will be benchmarked against the well established GNU Go, an open-source Go game that provides a consistent level of challenge. The goal is for SapioGo to beat GNU Go at least some small percentage of the time. Although other open-source Go games are available, GNU Go provides a consistent level of difficulty, making it easier to monitor SapioGo's progress.

For more information about GNU Go, or to download a free copy of the source code, visit the following website:

`http://www.gnu.org/software/gnugo/gnugo.html`

4.2 Software Requirements

SapioGo will make use of open-source software for everything from its development to its end user interface. This should allow SapioGo to be easily ported to platforms other than those directly supported.

4.2.1 Development Environment

SapioGo will be implemented in C++ using the GNU C++ compiler and will be targeted for a UNIX platform, such as FreeBSD 5.1, as well as for Microsoft Windows XP/2000. The source code is written in strict ANSI C++ and could be easily compiled on any other UNIX or LINUX platform.

4.2.2 Network Protocol

SapioGo will make use of the Go Modem Protocol (GMP) library, created by David Fotland. This library implements the standard JCGA Go Communication Protocol in C and handles all communication and interaction with other Go game clients.

The GMP library, along with sample Go applications using the library, are available for download from the cgoban website, listed below.

4.2.3 User Interface

SapioGo will make use of an existing Go interface program known as cgoban, version 1.9, which will provide a method of initiating new games, playing them, saving them, and restoring saved games. The more recent versions of cgoban do not provide a method for interacting with other automated Go clients, and thus SapioGo will not work with them.

Version 1.9 of cgoban can be downloaded freely at the following website:

`http://www.igoweb.org/~wms/comp/cgoban/`

4.3 Hardware Requirements

SapioGo will be able to run smoothly on modest hardware by today's standards. Since SapioGo performs a massive amount of floating-point computations when evaluating the neural network, a computer with a faster processor should be used if available.

Processor:	1.2 GHz Intel Pentium® III or comparable
RAM:	256 MB (preferably 512 MB)
Hard Drive Space:	500 MB
Peripherals:	Mouse, Keyboard
Display:	800x600 16-bit color display

References

[1] Creavier, Daniel. *AI: The Tumultuous History of the Search for Artificial Intelligence*. Basic Books, New York, New York, 1993, ISBN 0-465-02997-3.

[2] DeLoura, Mark. *Game Programming Gems*. Charles River Media, Inc., Rockland, Massachusetts, 2000, ISBN 1-58450-049-2.

[3] Iwamoto, Kaoru. *GO for Beginners*. Pantheon Books, New York, New York, 1976, ISBN 0-394-73331-2.

[4] LaMothe, André. *A Neural-Net Primer*. Charles RiverMedia, Inc., Rockland, Massachusetts, 2000.

[5] Lubberts, Alex. *Co-Evolving a Go-Playing Neural Network*. The University of Texas at Austin, 2001.

[6] Mitchell, Tom. *Machine Learning*. McGraw-Hill, Boston, Massachusetts, 1991, ISBN 0-07-042807-7.

[7] Moriarty, David E. *Efficient Reinforcement Learning through Symbiotic Evolution*. Kluwer Academic Publishers, Boston, Massachusettes, 1996.

[8] Richards, Norman. *Evolving Neural Networks to Play Go*. The University of Texas at Austin, 1996.

[9] Sharkey, Amanda. *Combining Artificial Neural Nets*. Springer, Great Britain, 1999, ISBN 1-85233-004-X.