CS 4701 AI practicum

Team Name: CC+

Team Members: Ningchuan Wan, William Moseson, Swinburn Miranda

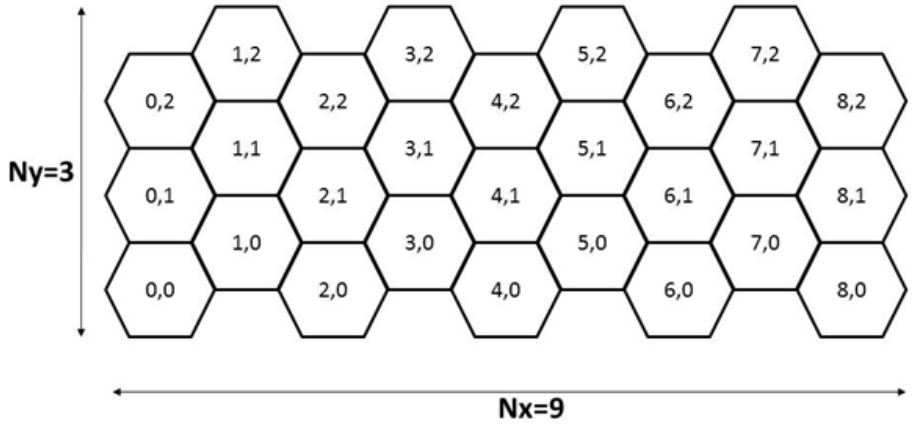
Problem Statement and Motivation:

The problem statement given is to develop an artificial Chinese checker's player which is capable of winning against other artificial as well as human players.
The motivation behind the project is the experience gained in the field of Artificial intelligence and the bragging rights achieved in beating other players.

I/O Specification

The program is given the initial board state containing occupied and unoccupied positions. After each round the server gives, as input, the last move performed. Our program also requests a desired move to the server as output.

The communication protocol is as follows



Players communicate via stdin and stdout with a game server, which enforces the rules of the game and takes care of obtaining the opponent's move and passing it to the player. The communication protocol is as follows:

Input

0 = Game initialization (followed by 3+Nx*Ny integers)
Total time available for the game (seconds) on one core
Board size (Nx, Ny)

List of Nx*Ny cell status numbers starting at (0,0) then (1,0) then (2,0) ... (Nx-1,0) then (0,1), (1,1), ... (Nx-1,1) until last row (0,Ny-1) ... (Nx-1, Ny-1). Each status number can be one of the following:

0 = unoccupied cell
1 = occupied by you
2 = occupied by your opponent
3 = unavailable (not part of the board)

1 = Request move (followed by one integer)
Remaining time (secs)

2 = Opponent moved (followed by four integers)

Start cell
End cell

3 = Error (followed by one integer)
1=Illegal Move (no piece or not your piece at start position)
2=Illegal Move (end/intermediate position occupied or off board)
3=Illegal Move (bad hop: not adjacent, or hopping over unoccupied cell)
4=Time out

4=End game

Output

0=Move (followed by 3+2N integers)
Number of hops (N)
Start cell coordinate
If N>1, list coordinates of N-1 intermediate cells
End cell coordinate

Background Reading

http://www.whitneybabcock-mcconnell.com/portfolio/Chinese%20Checkers.htm
This tells us every step about the developed algorithm for the Chinese checkers AI. He specifies detailed rules in comparison to other generalized approaches.

A Comparison of *Algorithms* for Multi-Player Games
It's a general algorithm for multi-player games with Chinese checkers as an example.

http://hem.passagen.se/baolan/release/china.pdf
Detailed description of the implementation of a search algorithm for the Chinese checkers game.

http://www.cs.northwestern.edu/~agupta/_projects/chinese_checkers/web/index.html
Lists and describes five different types of heuristics that are used for Chinese checkers. Also gives some discussion on optimization strategies.

General Approach

The basic building block for prediction of the next move would be determined by Minimax algorithm. The cost for each edge would be determined using heuristics such as: Distance of the piece from the goal, Potential travel distance, avoiding stragglers.

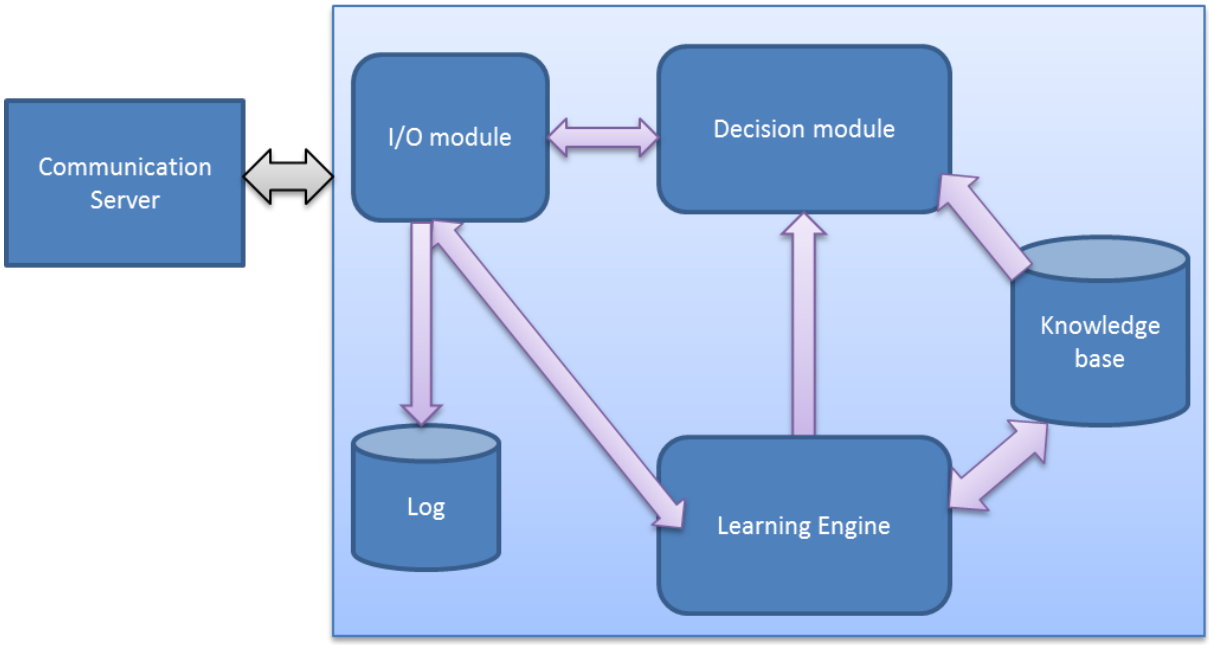
"Where is the AI"

The main AI components of the project would be the usage of Minimax algorithm using optimal heuristics to help determine the next optimal move. At the same time Machine learning would be used by inspecting past game and on going game plays to determine which player has the upper hand and hence respond accordingly.

CS 4701 AI practicum

Team CC+

System Architecture and work plan



I/O Module: Used for reading and writing out to and from the server and parsing the new state in the system for computation

Decision Module: It will implement the minimax algorithm and help pick the next optimal move.

Learning engine: It continuously observes the on going states and compares it with historical data to update the knowledge base and influence the decision modules.

Knowledge Base: Information derived from past games.

Log: Maintains a log of all moves made by the opponent and us along with information computed about the heuristic.

Data sources

Potential data sources are game plays which involve

- The bot playing with itself
- Other opponents who we chose to collaborate with.
- The dumb bot provided by the TA.

Evaluation Plans

1. Metrics:
    a. Time it takes to make a move
    b. Rate of winning matches
2. Simple Test:
    a. Win against the random move AI

3. Ultimate Test:
    a. Win against people
    b. Win against all other AI bots in tournament

Schedule

Week	Dates	Milestones
1	24th September	Class structure
2	1st October	I/O module
3	8th October	Basic Decision module with no learning or Knowledge base
4	15th October	Basic Learning engine
5	22nd October	Interface learning engine and decision module.
6	29th October	Begin build of knowledge base
7	5th November	Use knowledge base in decision module
8	12th November	Integration testing
9	19th November	Continue training the system
10 – last	29th November	Perfect the system + Project report