CS44 Project Proposal

Team member: Qi GU, Zhifei Song

1, Project Name:

Gomoku (Five-in-a-row) Game

2, Problem Definition:

Gomoku is an abstract strategy board game. It is traditionally played with go pieces on a go board, but the standard board for Gomoku is a little smaller (15*15 intersections) compared with a standard go board. The game has two kinds of pieces: black and white, black plays first, and players alternate in placing a piece of their color on an empty intersection. The winner is the first player to get an unbroken row of five stones horizontally, vertically, or diagonally.

In this project, we will incrementally build the game with an intelligent computer player, including following steps (we will explain each step more clearly in Approach section):

(1) Basic game building-up, after finishing this part, we can play Gomoku with other people.(Milestone1: finished before Feb 28)

(2) Computer AI design and implementation, after finishing this part, we can play Gomoku with computer, though the computer may not be very “intelligent”.
(Milestone2: finished before Mar 3)

(3) Using different strategies to improve the Computer’s AI for Gomoku.
(Milestone3: finished before Mar 6)

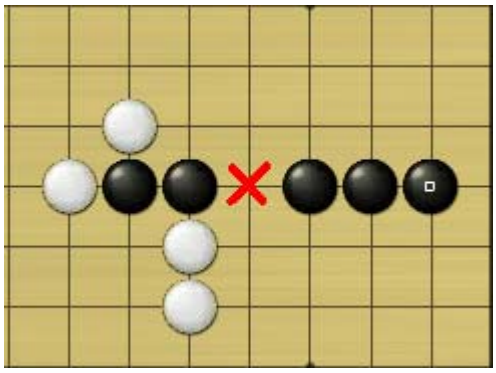
(4) Graphic user interface implementation. (Milestone4: finished before Mar 8):

(5) Final testing, finish all document (finished before Mar 13)

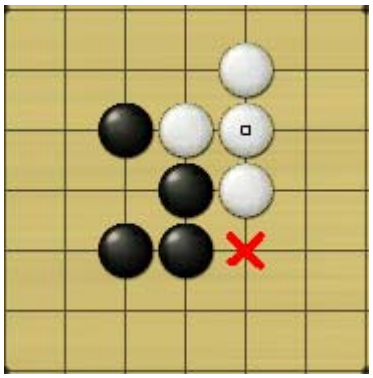
3, Project Approach.

Each step above can be splited further.

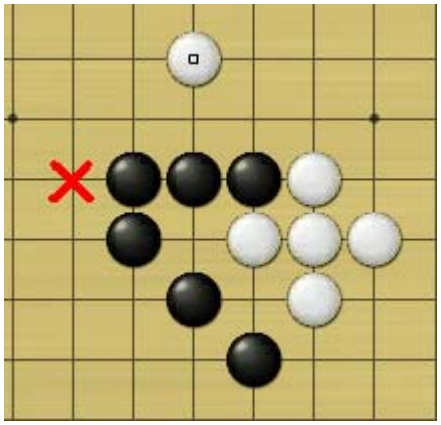
- Basic game building-up
  (1) Encoding: Defining the data structure for the Gomoku board. Decide how to display the board and pieces.
  (2) Initial state: function to start a new game or reset the game. Reading the initial state from a file.
  (3) Successor Function: including legal_moves function, do_move function and undo_move function.
  (4) Terminal States: There are three terminal states, black win, white win or draw. The black win can be achieved by making exact 5 black pieces in a row horizontally, vertically or diagonally; the white win can be achieved by 5 white pieces in a row or by black’s forbidden moving:
    a) Overline (more than 5 black pieces in a row),



b) Three-three (have two live three)



c) Four-four (live four or sleep four). The draw state is when the board is full, none of above states has been achieved.



(5) Command Line Interface: give out some game options: new_game, save_game, load_game and quit. After choosing whether to start a new game or load a game, we have following game options: Human vs. Human, Human vs. Computer.

- Basic Computer AI design and implementation

(1) Utility function and evaluation function: to guide the minimax search.

We consider black takes the role of Max in the minimax search:

a) If the black stones can form 5-in-a-row (without the forbidden moves mentioned above), the u(s) takes 100000 point, if the white can form 5-in-a-row, the u(s) takes -100000 points.

b) If there is live four, double sleep four or one sleep four and live three for

black, the u(s) takes 10000points; for white, u(s) takes -10000 points.

c) If there is double live three for white (double live three will kill black!), the u(s) takes -5000points
d) If there is one sleep 3 and one live 3 for black, u(s) = 1000 points, for white, u(s) = -1000points
e) If there is one sleep 4 for black, u(s) = 500 points, for white, u(s) = -500 points.
f) If there is one live 3 for black, u(s) = 200 points, for white, u(s) = -200 points.
g) If there is double live 2 for black, u(s) = 100 points, for white, u(s) = -100 points.
h) If there is one sleep 3 for black, u(s) = 50 points, for white, u(s) = -50 points.
i) If there is double sleep 2 for black, u(s) = 10 points, for white, u(s) = -10 points.
j) If there is one live 2 for black, u(s) = 5 points, for white, u(s) = -5 points.
k) If there is one sleep 2 for black, u(s) = 3 points, for white, u(s) = -3 points.
l) Other cases and draw states should return u(s) = 0 points.

(2) Minimax search.

- Using different strategies to improve the Computer's AI for Gomoku.
  (1) alpha-beta pruning.
  (2) Transposition table.
  (3) Ordering the successors and search the high-order successors further than others.
  (4) Threat Space Search，A winning threat sequence consists of threats. A threat is a live four, sleep four or the live three. This search will concentrate on the space of all threats. [2].
- Graphic user interface implementation

4, Our Project Goal and Test Cases:

Goal: Basically, our project goal is running the game gracefully, it should achieve our declared approach and the computer player do not make a stupid mistake during playing the game with the human player, the so-called stupid mistake is the computer does not try to take a win or block the human player. Furthermore, we expect our computer AI wouldn't lose to a normal human player in at least 30 plies. We may suppose Afra and Lu both belong to the normal human player :-)

Test Cases:

Initial test includes making sure the program work correctly and record the failed work so far.

Specific testing case will be developed and reported on the project updating

report.

5, Task Assignment.
Qi Gu: Team leader and main programmer, AI algorithm design.
Zhifei Song: Test cases design, write document and GUI implementation.

6, References:
[1] Stuart Russell, Peter Norvig, Artificial Intelligence A modern approach 2nd edition. Pearson Education, Inc.,2003.
[2]L.V.Allis, H.J.van den Herik, Huntjens, Go-Moku and Threat-Space Search.
[3]http://en.wikipedia.org/wiki/Gomoku
[4]Further Discussion with our professor Afra Zomorodian.