CS 396 Project 1

Gomoku

BY: Yanzhou Huang

“Gomoku” is a traditional game which have 15\*15 board. A player need to gets an unbroken row of five stones horizontally, vertically, or diagonally to win the games. Also it also the player uses some strategy such as Blocking the opponent, uses some trick step to hide the real intentions. In the rule, the black stone is always go first.



***Requirements***

This project requires that user can choose either player vs player and player vs com. Also the game allow user start the new game at any time. In this game, the player must be able to placing a stone by simply press on an open spot on the board and does not require the user to type in anything. And the game will show some message dialog to Recognized a win (5 in row) and a tie (the board is filled up). when is win and tie happen. User will not allow to do more moves until the new game begin. In the player vs com, the computer cannot just play randomly. It must have some strategy, such as blocking opponent and willing to get an row of five stone.

***Design***



The Language in this game is JAVA. Use JAVA swing(Java Platform SE 7) to design the interface, button, background and draw the graphic.

***Interface***

The interface is in the “MainFrame” class. JLabel is used to make some simple background and title. JButton is used to create some control button. ActionEvent is used to swtich the interface to the main board. Left Mouse press is the function action in this game.

***gameplay***

The “ChessBoard” class is the main logic class. In this class, array is used to hold the chess, Graphics G is used to draw the board, chess, mouselistener is used to monitor the mouse pressed on the board. Actionlistener is used to monitor the button. This class also has methods to return whether a game is win or draw and managing the chess placement.

The count is used to check the amount of the stone on the board, the board is 15\*15 which mean it can be filled by 225 stones on the board for both side. So if the count meets the maximum, which mean the game is a tie. Win a game or not

is determining by a five in unbroken row. Used four loop to evaluate the player who hold the white is win, or black is win. or win the computer or the computer win the game.

“Judge” class is used 8 possible connection way to determining the connected stone.



“chess” class is used to create some information such as direction of the connection, the level of the stone placement. And the turn for player or computer

***Computer***

“MyList” class is used to store the chess and find the best placement on the board.

The computer logic is programmed in the “AI” class. Use level to distinguish the species type of the chess. Use if else statement to judge the priority. Use 8 case (represent 8 direction) to evaluate the degree for the placement.

For the AI strategy,

First step, Based on both situation, assumption giving both one point, and to see the change for the board situation. If it could not connected to 4, then check both can be connected to 3.

Second step, according the results of the first step, combine the consequence for each point. if it can be connected to 3 without block on two side, or can be connected to 4.

Third step, Base on the rules, rank the results, and select the stone placement (offensive act or defensive act) depends on how possible the empty space can be used to form a ‘five’ if a piece is set on it.



Before each step need to search the spot have the chess already placement or not.

If don’t meet the situation, then choose the random placement on the board.

When place the chess on the spot make sure there is no chess.

***Bugs***

It may have some bug when you clicked the board without intention. So when this happen, it will show error message dialog then remove the mouseListener which mean the player could not make more move instead of start a new game.

***Limitations***

Eclipse is the most common way to run this program.

***Unimplemented Features***

The features are not implemented in this project;

1. **Step** function. Show step for the players and computer. It may similar with the count method.
2. **Undo** function. Sometime the player may place a chess by mistake or do not want to place that spot.
3. Beautify the GUI and chessboard. Right now just have the simple background and simple function. Later I can add a menu bar for the interface and add the background for the chessboard.
4. **AI** function. Now the AI just can do some best search and move under five steps. That the AI will not have the great strategy, which mean when the AI have 3 stone connect, it will try to connect to 5, and ignore the stone that player place. Later I may rank the move again, before each move, make sure the AI search the board first, that chose the suitable strategy. May be use some alpha-beta algorithm to create the AI later.

***Acknowledgements***

The tutorials of creating a GUI and mouselistener with JSwing

<https://docs.oracle.com/javase/tutorial/uiswing/>



https://docs.oracle.com/javase/tutorial/uiswing/events/mouselistener.html

video on youtube about the Action and Listener for tictactoe

<https://www.youtube.com/watch?v=Db3cC5iPrOM>

video on youtube about gomoku

<https://www.youtube.com/watch?v=lebCk5SLnrU>

video on youtube about five in row

<https://www.youtube.com/watch?v=UYI_gr6BUCM>

The source code of AI part online create by javascript

<https://github.com/yyjhao/HTML5-Gomoku/blob/master/js/ai-worker.js>

image download online

***Screen shot***

