**Report on Project: Implementing Go-Moku (five-in-row) Game**

**Team members:**

**Piran Caci**

**Simen Sverdrup-Thygeson**

**Ojin Kim**

**Abstract**

This report provides a brief overview of the game five-in-row, also known as Go-Moku (from Japanese: five-stones), the basic structure of our program, including the single-player and multi-player mode. The multi-player mode consists of two player giving input from the command line, whereas the single-player mode is a more complicated one, since we had to implement a computer-based opponent player, which takes reasonable decisions. In the single-player mode the player is playing against the computer. The program doesn’t provide a GUI (Graphical User Interface) and will only be played from giving input in the command line.

**Introduction**

The five-in-row game is a two-player abstract strategy board game often played with Go board and pieces. Its rules are simple, but they lead to a highly complex game. The board consists of n horizontal lines and n vertical lines, where two players Black and White take turns to mark their own color on one of the empty squares. The board is more often seen in sizes of 15 x 15 or 19 by 19. Five-in-row game is a game with long history. The Black player is the one who shall start the game and is believed that the first player has an advantage of winning the game. For that purpose, we choose to set the size of the board by 15 x 15, which is believed to reduce the advantage of moving first. Our board has rows marked with alphabetically ordered capital letters, whereas colons are marked with numbers. In our program, since we don’t provide Graphical User Interface, our players are marked with O for the Black player and X for the White one. These are the rules of the game; once a marker is placed on the board (in our case O for the black player starting the game ), it can’t be moved to another position later. A marker can’t be taken over by another player, either. The player, who creates a line of five consecutive markers of his color/symbol horizontally, vertically or diagonally, wins the game. A consecutive line of 6 markers can be theoretically created but will not be counted for the player who creates it, while the opponent player will not have any restriction. If no one creates five-in-row before the board is completely filled, the game is drawn.

**Description of the program**

The program is based on the following classes;

* Board Class, which stands for creating a board in size by 15 X 15. Part of the Board class are two fundamental public methods;

***public void displayBoard();*** this method is for displaying the actual board every time a player’s turn comes.

***public void insertRock(String position, char rock);*** this method is for allowing each player to place a rock in the board. In our case a symbol X/O. The method will take two parameters; the position of the rock and the rock itself.

* Move Class, which contains numerous public methods;

**public boolean checkEmpty(int a, int b);** this method checks whether the board position, in which the player tends to place its rock, is free. It takes to parameters a & b, which represent respectively the coordinates of the position, where the move should be done.

**Public Boolean withinBoard(int a, int b);** this method checks whether the position coordinates are within the board. It takes also two parameters, which represent the coordinates of the position.

**Public Boolean legalMove(String position);** this method checks whether the intended move is legal. Basically it checks whether the coordinates of the positions are within the board and at the same time free. It takes a string parameter consisting of the position in the format (i.e. a12).

**Public Boolean gameFinished(String position);** this method checks whether the game is finished. Likely the above methods do, it also take a string position as parameter.

🡪 Singleplayer Class, which is for the Single-player mode of the game. It inherits the Board class and creates a Move & Ranking object as well.

**Public void play();** this is the only method of this class. Inside the method is checked whether the move is legal, game is finished, the board is displayed and the rock is inserted by calling the certain methods of the Board & Move classes.

🡪 Multiplayer Class, which is for the multi-player mode of the game. It creates objects from the Board, Move & Ranking classes. Similar to the Singleplayer class also provides following method;

Public void play(); this method works almost exacltly as the play method of the Singleplayer class.

🡪 Ranking Class, which provides rankings and scores of the players after the game is finished.

It provides following methods:

**Public void displayscore();** this method gets access to a .txt file named highscore, which provides the scores of the players and displays them.

**Public void appendScore(String player1, String player2, int moves);** this methods gets the scores of each player and the scores are based on the number of moves that led the winning player to win the game. It gets three parameters as reference; Two string parameters with player’s names and the third integer parameter, which represents the number of moves.

**Public void resetScore();** this method works by deleting the file of the previous highscores and creates a new one.