First assignment

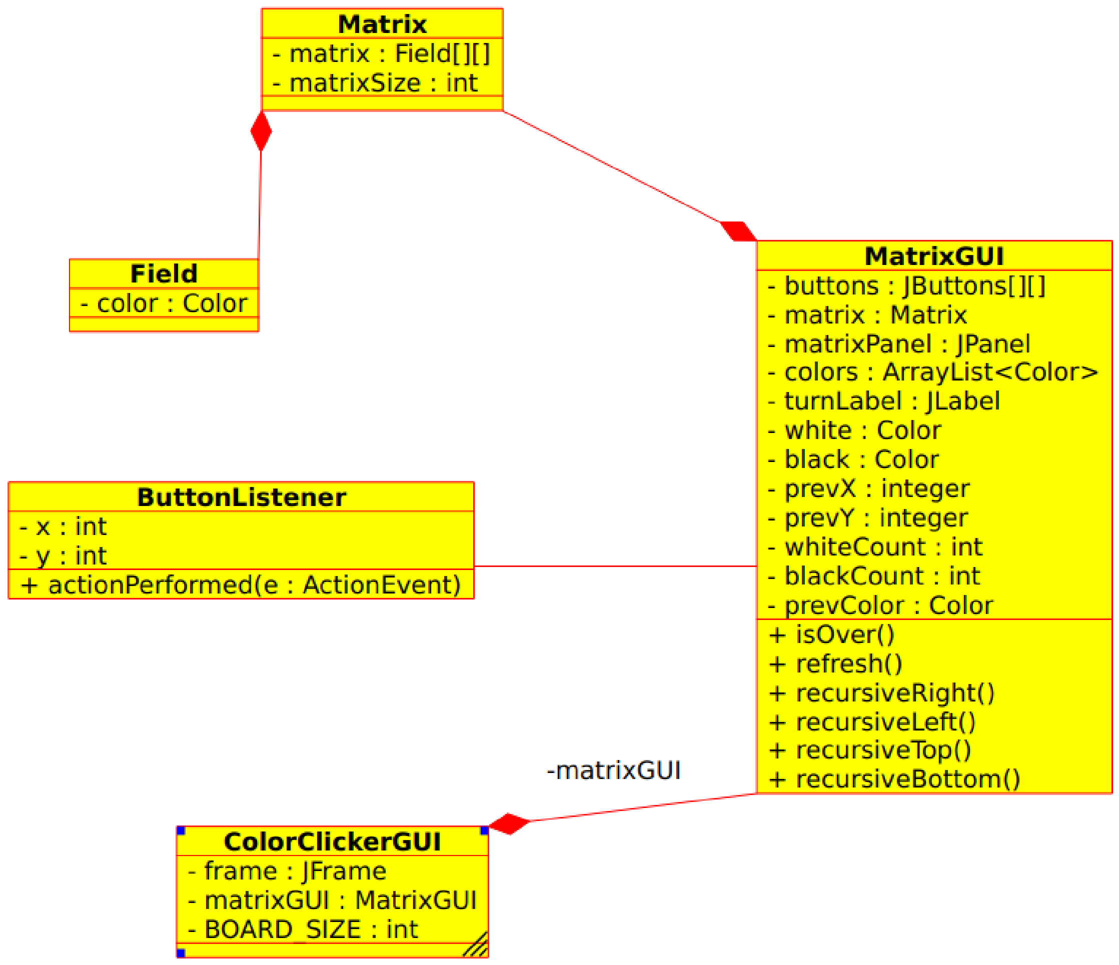
Student name: Khongorzul Khenchbish
Student id: r8hp78

Description:

2. Pebble

Pebble is a two-player game, played on a board consists of n x n fields. Initially, n white and n black pebbles are placed on the board randomly. Each color belongs to only one player. The players take turns choosing one of its pebble, and then move it horizontally or vertically. The movement also affects the neighbouring pebbles in the direction (the pebble on the edge falls off). The objective of the game is to push out as much pebbles of the opponent from the board as we can, within a given number of turns (5n). A player wins, if he has more pebbles on the board at the end than his opponent. The game is draw, if they have the same number of pebbles on the board. Implement this game, and let the board size be selectable (3x3, 4x4, 6x6 → turns are 15, 20, 30). The game should recognize if it is ended, and it has to show the name of the winner in a message box (if the game is not ended with draw), and automatically begin a new game.

Class diagram:



Short description of each methods:

Matrix GUI class:

1. ***isRefresh()***
   This function is called After each move and compares the number of pebbles players left with and returns true if either one of the sides wins or both draw. Both players draw when they have the same number of pebbles left with. Pop up windows should appear in all of the above three cases. If the players don't run out of turns then it updates the turns.

2. ***startGame()***
   The two players start playing in turns one by one and the game finishes only when “either one of them loses all the pebbles on the boards” or we “run out of the number of turns”. In each turn, the player will first Attack() to the opponent with random move,

however the opponent side won’t be pushed out of the board until the Eliminate() function is executed. The pushing out will happen only if the opponent is on the edge of the board and the player moves to that position.

3. ***isOver()***
   It counts the number of pebbles players left with and the game is over when either one of the players runs out of pebbles or they reach the total number of turns.

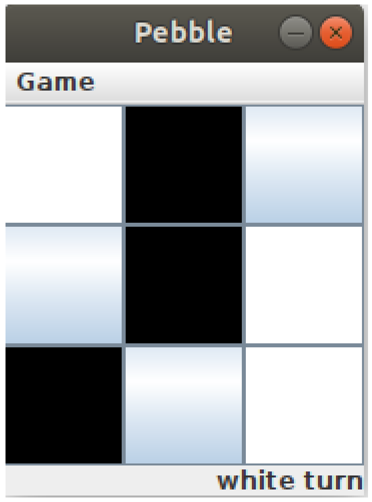
4. ***recursiveRight, Left, Top, Bottom()***
   On each move, this function calculates the position for each pebble that was pushed in the specific direction until it reaches the edge. If the opponent’s pebble is on the edge, it will be pushed out of the board.

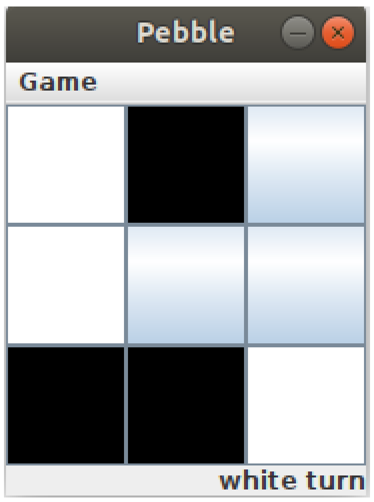
ButtonLIstener class:

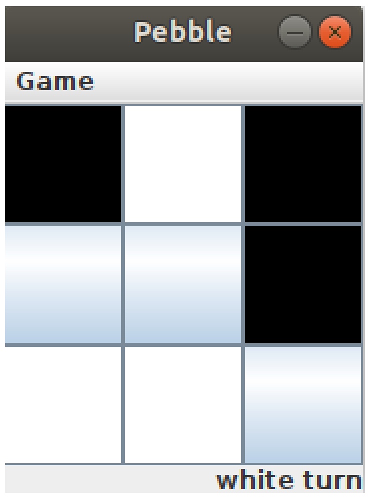
1. ***ActionListener()***
   It checks three cases.
   1. We don’t know where the pebble will be, we only know it’s current position. In this case, we don’t do anything.
   2. We know where the next position is. Here we will determine if the pebble wants to move to up, down, left or right then we calculate the move and affected pebble positions.
   3. The move is not allowed. It contains cases where the player wants to move diagonally, try to attack itself accidentally, try to move missing spaces and pebbles in between.

Test cases:

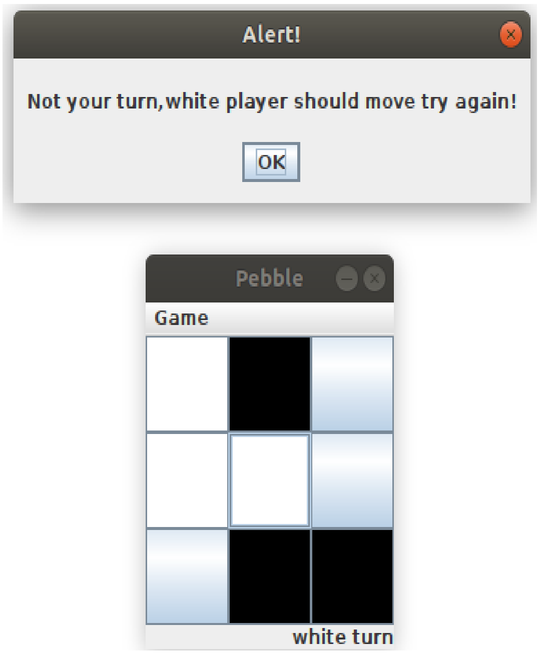
1. Checked if the pebbles are placed randomly on the start of a game.



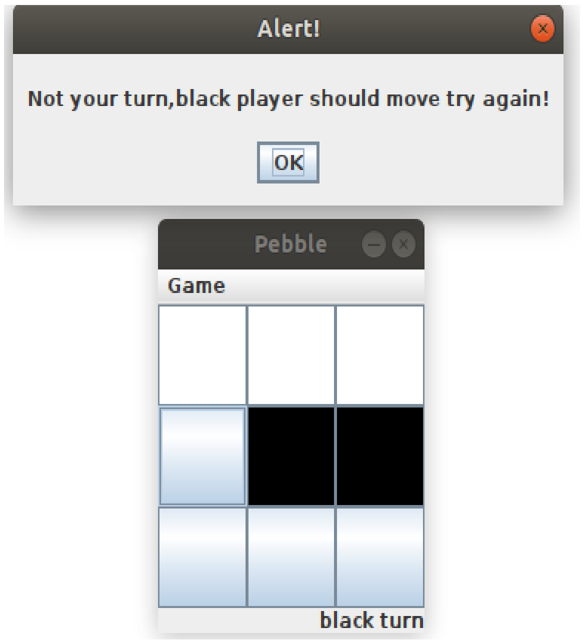




2. If it is not the player's turn, then the pop up shows the alert “It’s not your turn.”

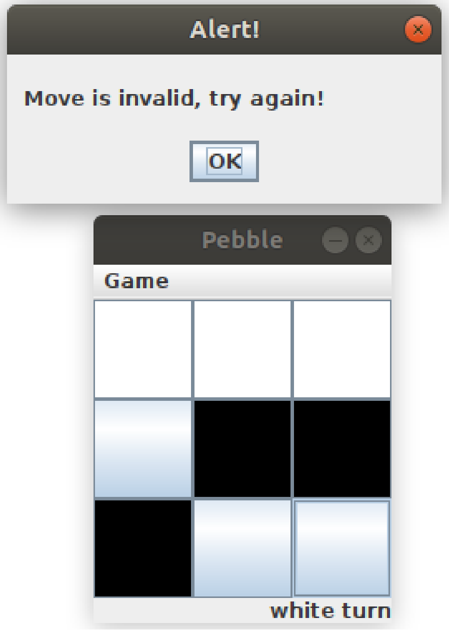

Alert!

Not your turn,white player should move try again!


Alert!

Not your turn,black player should move try again!

3. Same as the previous, the application returns an alert on diagonal and invalid moves.


Alert!

Move is invalid, try again!