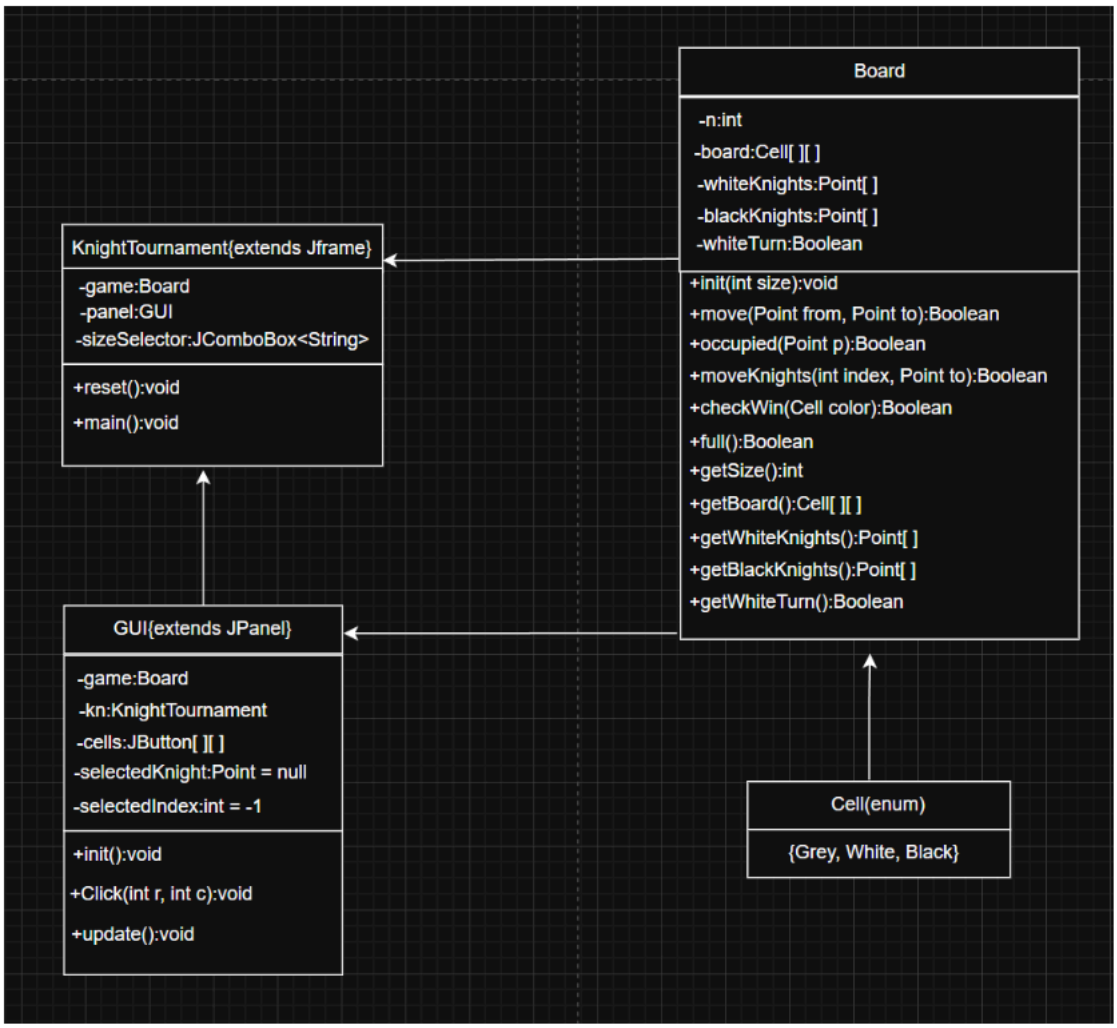
8.Knight tournament

Description:

This two-player game is played on a board consists of n x n fields. Initially, two white and two black knights are placed at the corners of the board (knights of the same color are placed at the opposite corners). Players step alternately, and knights can move only in an L shape, like in chess. The board is initially grey, but after each step, the visited places are colored with the color of the visitor knight (the previous color of the field doesn't matter). A player wins, if there are 4 adjacent fields (horizontally, vertically, or diagonally) which colored to player's color. The game ends, when there are no more grey fields. Implement this game, and let the board size be selectable (4x4, 6x6, 8x8). The game should recognize if it is ended, and it has to show in a message box which player won. After this, a new game should be started automatically.

Class diagram:



Description of methods:

Board:

init(): Set every Cell in the matrix to Grey, two white and two black knights are placed at the corners of the board. And the whiteTurn = true so white knigts can move first.

move(): Get the number of horizontally and vertically moved steps, if that correspond to the L shape, return true;

occupied(): Return true if the point is occupied by a knight.

moveKnight(): If whiteTurn = true, we control the white knights and vice versa. If the target point is occupied or the movement isn't like L shape, return false. If it's okay, move the knight to the target point and paint the point to corresponding color, continue to next turn.

checkWin(): If there are 4 adjacent fields (horizontally, vertically, or diagonally) which colored to white or black, return true;

full(): If every cell is white or black, return true.

getSize(): Return size of the board.

getBoard(): Return the matrix.

getWhiteKnights(): Return the list of white knights.

getBlackKnights(): Return the list of black knights.

getWhiteTurn(): Return it's whiteTurn or not.

GUI:

init(): Make a Jbutton matrix. And for each place, assign a button, add a handler for buttons which controls what is going to happend after click. add() every button so they can be visible.

click(): If we did not select a knight, based on given point we select the knight on that point. If we already selected a knight, so we suppose to move it for the next click, If the movement is like L shape then it'll be executed, otherwise we don't do anything. And after every successful movement we check if a color wins or not or it's a draw.

update(): For each button that is occupied we set it to black or white, if it's not occupied then grey. And we use letters to represent knights.

KnightTournament:

reset(): Restart the game with selected size.

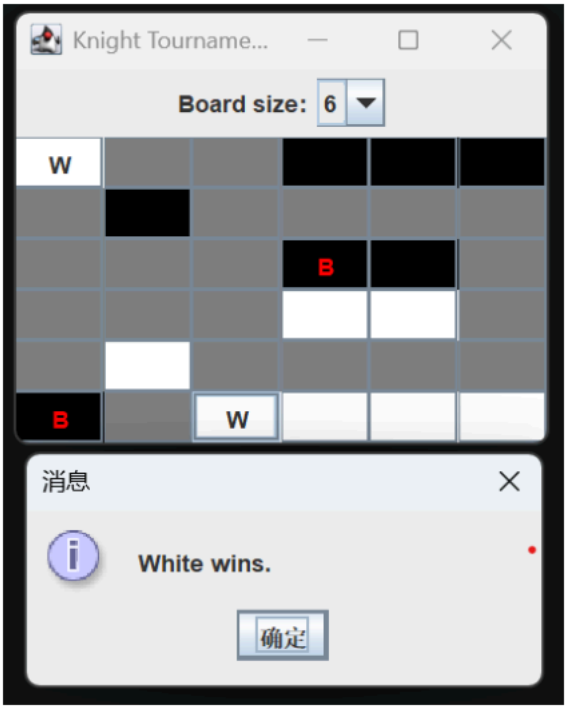
Events and event handlers:

Click on the buttons will call the handler click().

Chose a size from the drop-down list will call the handler reset().

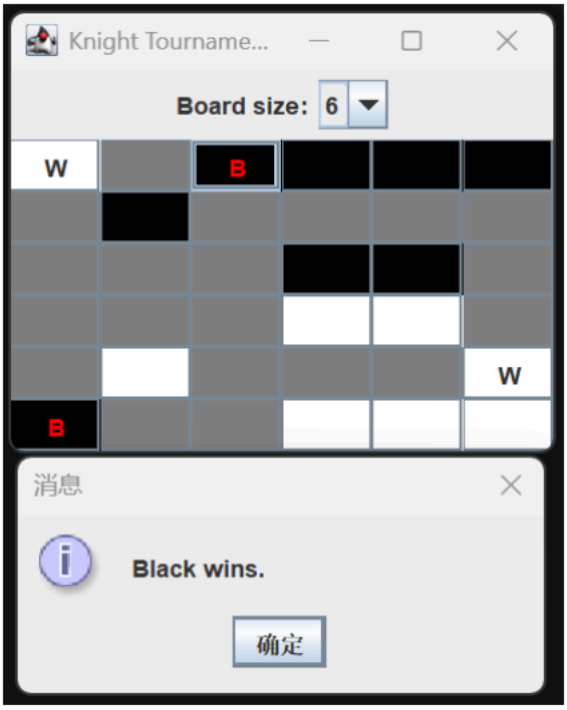
Tests:

Case1:



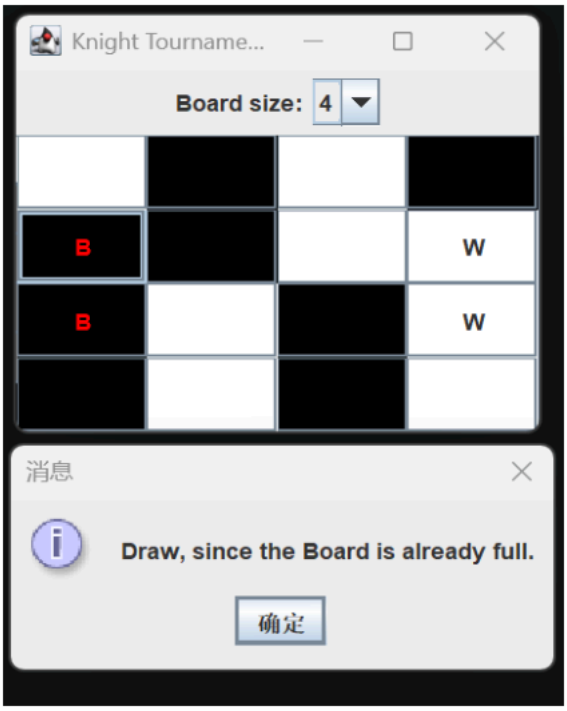
Result: 4 horizontal adjacent fields for white, white wins.

Case2:



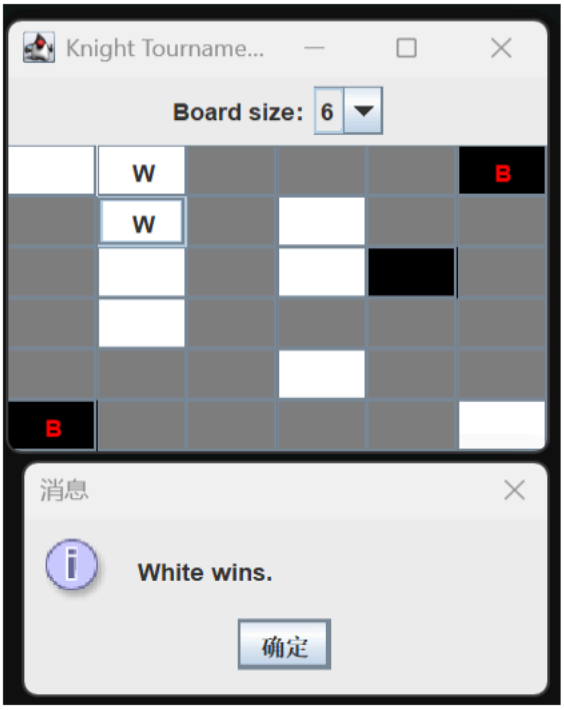
Result: 4 horizontal adjacent fields for black, black wins.

Case3:



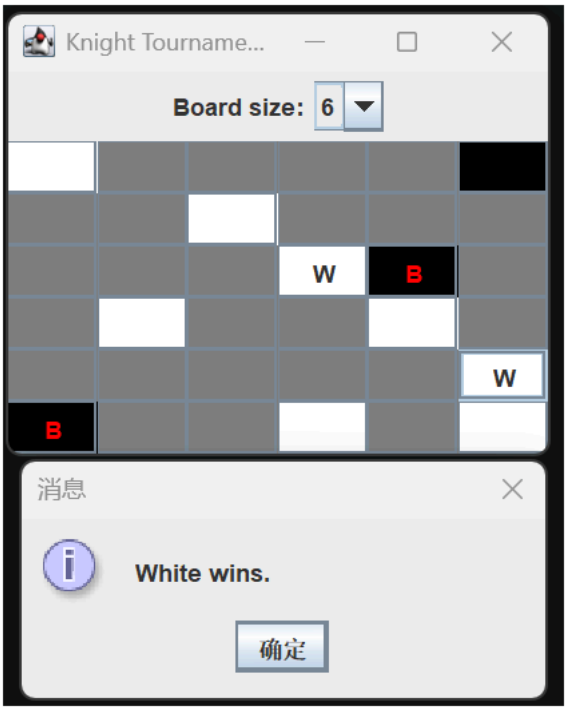
Result: Draw, since the board is already full.

Case4:



Result: 4 vertical adjacent fields for white, white wins.

Case5:



Result: 4 diagonal adjacent fields for white, white wins.