Documentation

Project Description:

Hunting

Hunting is a two-player game, played on a board consists of n x n fields, where the first player (call him fugitive) tries to run away, while the second player (the hunter) tries to capture him/her. Initially, the character of the fugitive is at the center of the board, while the hunter has four characters (one at each corner). The players take turns moving their character (hunter can choose from 4) 1 step on the board (they cannot step on each others character). The objective of the hunter is to surround the fugitive in at most 4n steps, so it won't be able to move. Implement this game, and let the board size be selectable (3x3, 5x5, 7x7 → turns are 12, 20, 28). 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.

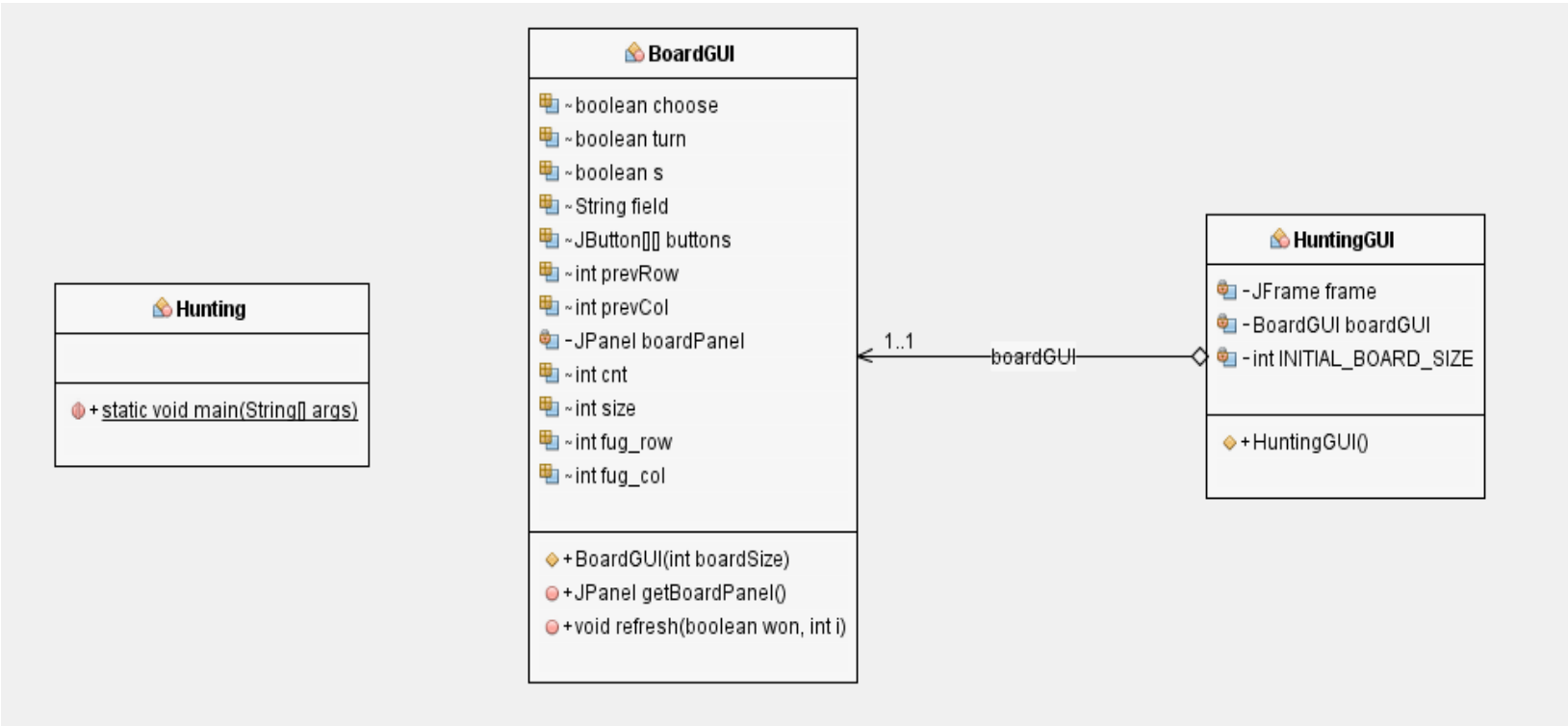
User Doc:

There are 4 hunters and 1 fugitive, the task of hunters is to catch the fugitive in such a way that he should not have any chance to scape, basically they have to block all the ways of fugitive.

The default board size is 3x3, but you can change and this game can be played in 5x5 or 7x7 size board as well.

If hunters catch fugitive in 4 * board size steps, then hunters win, otherwise fugitive wins the game.

UML Diagram:



Implemented Methods:

```
public BoardGUI(int boardSize) {

    size = 4 * boardSize;
    boardPanel = new JPanel();
    boardPanel.setLayout(new GridLayout(boardSize, boardSize));
    JButton[][] button = new JButton[boardSize][boardSize];
    //boolean choose = true; //???
    int pos = 1;///??

    /**
     * ***To set the buttons in array***
     */
    for (int i = 0; i < button.length; i++) {

        for (int j = 0; j < button[i].length; j++) {
            String player = "";

            //For Hunter
            if (i == 0 && j == 0 || i == 0 && j == button.length - 1 || i == button.length - 1 && j == 0
|| i == button.length - 1 && j == button.length - 1) {

                player = "H";
                player = player + new Integer(pos).toString();
                pos++;
```

```
        }

        //For Fugitive
        if (i == button.length / 2 && j == button[i].length / 2) {
            player = "F";
            fug_row = i;
            fug_col = j;
        }

        JButton g = new JButton(player); //button g

        g.setBackground(Color.YELLOW);

        button[i][j] = g; /// elements m
        button[i][j].addActionListener(new MoveActionListener(i, j));
    }
}



/**
 * ***To Put the buttons in grid***
*/
for (int i = 0; i < button.length; i++) {
    for (int j = 0; j < button[i].length; j++) {
        boardPanel.add(button[i][j]);
    }
}
```

```
    // to use outside of the class
    buttons = button;
  }
```

This method sets the buttons for hunters and fugitive and put them in the grid.

```
public void actionPerformed(ActionEvent a) {
        JButton click = (JButton) a.getSource();
        String select = click.getText();

        if (s == true) {
           if (choose == false) {
              if (turn == true) {
                 if (select.equals("H1") || select.equals("H2") || select.equals("H3") ||
select.equals("H4")) {
                    ///to store prev value
                    field = click.getText();
                    prevRow = x;
                    prevCol = y;
                    click.setText("");
                    choose = true;
                 } else {
                    System.out.println("Please select the hunter!");
                 }
```

```
    } else if (turn == false) {
        if (click.getText() == "F") {
            field = "F";
            prevRow = x;
            prevCol = y;
            click.setText(""); //??
            choose = true;
        } else {
            System.out.println("Please select the fugitive!");
        }
    }
} else if (choose == true) {
    if (turn == true) {
        ///for no player field
        if (click.getText() == "") {
            if(prevRow!=x || prevCol!=y){

            if (prevRow + 1 == x || prevRow - 1 == x || prevRow == x) {
                if (prevCol + 1 == y || prevCol - 1 == y || prevCol == y) {
                    click.setText(field);  ///for replacing player
                    turn = !turn;
                    choose = false;
                    field = "";
                    cnt++;
                }
            } // If hunter selects the filed which is already taken
            else {
```

```
                    System.out.println("Please select the correct place for hunter!");
                }
            }
        } else if (select.equals("H1") || select.equals("H2") || select.equals("H3") ||
select.equals("H4")) {

            buttons[prevRow][prevCol].setText(field);///replacing field to old place
            field = click.getText();
            click.setText("");
            prevRow = x;
            prevCol = y;

        } //If hunter does not select the nearby row or coloumn
        else {
            System.out.println("Please select the correct place for hunter!");
        }
    } else if (turn == false) {
        if (click.getText() == "") {
            if(prevRow!=x || prevCol!=y){
            if (prevRow + 1 == x || prevRow - 1 == x || prevRow == x) {
                if (prevCol + 1 == y || prevCol - 1 == y || prevCol == y) {
                    click.setText(field);
                    fug_row = x;
                    fug_col = y;
                    turn = !turn;
                    choose = false;
```

```
                    field = "";
                }
            } //If fugitive selects the filed which is already taken
            else {
                System.out.println("Please select the correct place for fugitive!");
            }
        }
        } //If hunter does not select the nearby row or coloumn
        else {
            System.out.println("Please select the correct place for fugitive!");
        }
    }

    if (cnt == size) {
        s = false;
        refresh(s, 1);
    }else if (fug_row == 0 && fug_col == 0) {
        if (buttons[fug_row][fug_col + 1].getText() != "" && buttons[fug_row +
1][fug_col].getText() != "" && buttons[fug_row + 1][fug_col + 1].getText() != "") {
            s = false;
            refresh(s, 2);
        }
    }
    else if (fug_row == buttons.length - 1 && fug_col == buttons[0].length - 1) {
        if (buttons[fug_row - 1][fug_col].getText() != "" && buttons[fug_row][fug_col -
1].getText() != "" && buttons[fug_row - 1][fug_col - 1].getText() != "") {
            s = false;
            refresh(s, 2);
```

```
                }
            }
            else if (fug_row == 0 && fug_col == buttons[0].length - 1) {
                if (buttons[fug_row][fug_col - 1].getText() != "" && buttons[fug_row + 1][fug_col].getText() != "" && buttons[fug_row + 1][fug_col - 1].getText() != "") {
                    s = false;
                    refresh(s, 2);
                }


            } else if (fug_row == buttons.length - 1 && fug_col == 0) {
                if (buttons[fug_row - 1][fug_col].getText() != "" && buttons[fug_row][fug_col + 1].getText() != "" && buttons[fug_row - 1][fug_col + 1].getText() != "") {
                    s = false;
                    refresh(s, 2);
                }
            }


        }
    }



  }
}
```

This method checks if the player 1 is choosing hunter 1st and place it to its right place or not, and 2nd player chooses the correct place for fugitive or not. Or its fugitive turns or hunters.

```
public JPanel getBoardPanel() {
        return boardPanel;
    }
```

This method returns the board panel in main.

```
public void refresh(boolean won, int i) {
        if (won == false) {

            /**
            * ***If Fugitive Wins***
            */
           if (i == 1) {
              JOptionPane.showMessageDialog(boardPanel,  "Fugitive won!", "Congratulations!",
                    JOptionPane.PLAIN_MESSAGE);

              HuntingGUI gui = new HuntingGUI();
           }
             /**
             * ***If Hunter Wins***
             */

           if (i == 2) {
              JOptionPane.showMessageDialog(boardPanel, "Hunter won!", "Congratulations!",
                    JOptionPane.PLAIN_MESSAGE);

              HuntingGUI gui = new HuntingGUI();
```

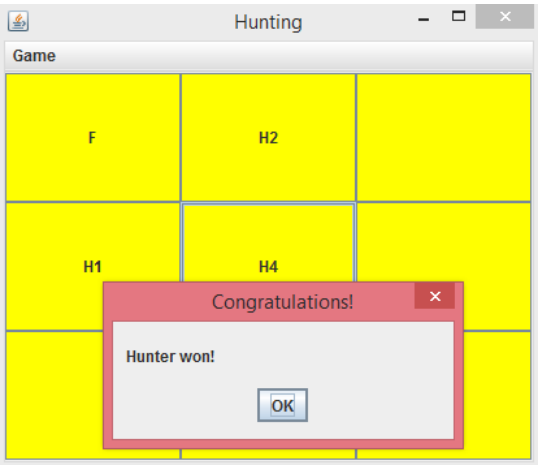
```
        }
    }
```

This method tells that who wins, hunter or fugitive and shows winner name as message dialogue.

Test Cases:

3x3

Hunter wins



Fugitive wins