```java
package project2;

import javax.swing.*;
import javax.swing.border.Border;
import javax.swing.border.LineBorder;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class SuperTicTacToePanel extends JPanel {

    private JButton[][] board;
    private Cell[][] iBoard;

    private JButton quitButton;
    private JButton undo;

    private ImageIcon xIcon;
    private ImageIcon oIcon;
    private ImageIcon emptyIcon;

    private SuperTicTacToeGame game;

    private int width;
    private int height;
    private int connections;
    private Cell starter;
    private boolean AI;

    public SuperTicTacToePanel() {
        this(3, 3, 3, Cell.X, false);
    }

    public SuperTicTacToePanel(int pHeight, int pWidth,
                        int pConnections, Cell pStarter, boolean pAI) {

        //resizes the image icons to the square size
        xIcon = new ImageIcon("./src/project2/x.jpg");
        Image imagex = xIcon.getImage();
        Image newimgx = imagex.getScaledInstance(60, 60,
                java.awt.Image.SCALE_SMOOTH);
        xIcon = new ImageIcon(newimgx);

        oIcon = new ImageIcon("./src/project2/o.jpg");
        Image imageo = oIcon.getImage();
        Image newimgo = imageo.getScaledInstance(70, 70,
                java.awt.Image.SCALE_SMOOTH);
        oIcon = new ImageIcon(newimgo);

        emptyIcon = new ImageIcon("./src/project2/empty.jpg");
        Image imagee = emptyIcon.getImage();
        Image newimge = imagee.getScaledInstance(60, 60,
                java.awt.Image.SCALE_SMOOTH);
        emptyIcon = new ImageIcon(newimge);

        JPanel bottom = new JPanel();
        JPanel center = new JPanel();

        // create game, listeners
        ButtonListener listener = new ButtonListener();

        quitButton = new JButton("quit");
        add (quitButton);
        quitButton.addActionListener(listener);

        undo = new JButton("undo");
        add(undo);
        undo.addActionListener(listener);

        width = pWidth;
        height = pHeight;
        connections = pConnections;
        starter = pStarter;
        AI = pAI;
        game = new SuperTicTacToeGame(height, width, connections,
                                        starter, AI);

        center.setLayout(new GridLayout(height,width,3,2));
        Dimension temp = new Dimension(60,60);
        board = new JButton[height][width];

        // add all the squares to the board
        for (int row = 0; row < height; row++)
            for (int col = 0; col < width; col++) {

                Border thickBorder = new LineBorder(Color.blue, 2);

                board[row][col] = new JButton ("", emptyIcon);
                board[row][col].setPreferredSize(temp);
                board[row][col].setBorder(thickBorder);

                board[row][col].addActionListener(listener);
                center.add(board[row][col]);
            }

        if (starter == Cell.O)
            game.AI();

        displayBoard();

        // add all to contentPane
        add (new JLabel("Super TicTacToe"), BorderLayout.NORTH);
        add (center, BorderLayout.CENTER);
        add (bottom, BorderLayout.SOUTH);
    }

    private void displayBoard() {
        iBoard = game.getBoard ();

        for (int r = 0; r < height; r++)
            for (int c = 0; c < width; c++) {
                if (iBoard[r][c] == Cell.O)
```

```java
                    board[r][c].setIcon(oIcon);
                else if (iBoard[r][c] == Cell.X)
                    board[r][c].setIcon(xIcon);
                else
                    board[r][c].setIcon(emptyIcon);
            }
    }

    private class ButtonListener implements ActionListener {

        public void actionPerformed(ActionEvent e) {
            if (e.getSource() == quitButton) {
                if (JOptionPane.showConfirmDialog(null,
                        "Are you sure you want to quit?", "YES", 2) == 0)
                    System.exit(0);
            }

            if (e.getSource() == undo){
                try {
                    game.undo();
                    if (AI)
                        game.undo();
                    displayBoard();
                } catch(IndexOutOfBoundsException er){
                    return;
                }
            }

            for (int r = 0; r < height; r++)
                for (int c = 0; c < width; c++)
                    if (board[r][c] == e.getSource()
                            && game.getBoard()[r][c] == Cell.EMPTY) {
                        game.select(r, c);
                        displayBoard();
                        if (AI && game.getGameStatus()
                                == GameStatus.IN_PROGRESS)
                            game.AI();
                    }
            displayBoard();

            if (game.getGameStatus() == GameStatus.X_WON) {
                JOptionPane.showMessageDialog(null,
                        "X won and O lost!\n The game will reset");
                game.reset();
            } else if (game.getGameStatus() == GameStatus.O_WON) {
                JOptionPane.showMessageDialog(null,
                        "O won and X lost!\n The game will reset");
                game.reset();
            } else if(game.getGameStatus() == GameStatus.CATS) {
                JOptionPane.showMessageDialog(null,
                        "Cats Game!\n The game will reset");
                game.reset();
            }
            displayBoard();
        }
    }
```

```
}

```