```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package fourplay;

import java.util.Observer;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.io.File;
import java.io.IOException;
import javax.imageio.ImageIO;
/**
 *
 * @author Lee Hudson   09092543
 * This class is the main board GUI. It contains many ColomnPanel objects
 * to show the game pieces.
 */
public class FPView implements Observer, ActionListener {

    private Dimension boardSize;
    private int noOfColomns;
    private int noOfRows;
    private JFrame board;
    private JPanel boardPanel,buttonPanel;
    private JButton reset,endGame,cpu;
    private ColomnPanel[] colomns;
    private JLabel player1Score,player2Score,player1Label,player2Label;
    private FPController gameController;
    private FPModel gameModel;

    public FPView(FPController gameController, FPModel gameModel){
        this.gameController=gameController;
        this.gameModel=gameModel;

        noOfColomns=gameModel.getNoOfColomns();
        noOfRows=gameModel.getNoOfRows();
        boardSize = new Dimension(noOfColomns*100,noOfRows*100);

        gameModel.addObserver(this);
        createBoard();
        gameController.setView(this);
    }

    /**
     * Sets up the GUI components
     */
    public void createBoard(){

        //Initialise GUI components
        board = new JFrame("FourPlay");
        boardPanel = new JPanel();
        buttonPanel = new JPanel();
        reset = new JButton("Reset");
```

```
        endGame = new JButton("End Game");
        cpu = new JButton("Activate CPU");
        colomns = new ColomnPanel[noOfColomns];

        //Set action listners for the buttons
        reset.addActionListener(this);
        endGame.addActionListener(this);
        cpu.addActionListener(this);

        //Set up main JFrame"board"
        board.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container contentPane = board.getContentPane();
        contentPane.setLayout(new BoxLayout(contentPane, BoxLayout.Y_AXIS));

        //Set up JPanel that contains the chips "boardPanel". The board panel
        //is made up of ColomnPanel objects.
        boardPanel.setLayout(new BoxLayout(boardPanel, BoxLayout.X_AXIS));
        for(int i=0; i < noOfColomns; i++ ){
            colomns[i]= new ColomnPanel(boardSize.width/noOfColomns,boardSize.height,noOfRows
            ,i,gameModel,this);
            boardPanel.add(colomns[i]);
        }

        //Set up the button JPanel "buttonPanel"
        buttonPanel.setLayout(new GridLayout(1,2));
        buttonPanel.add(reset);
        buttonPanel.add(endGame);
        buttonPanel.add(cpu);
        reset.setEnabled(false);
        endGame.setEnabled(false);

        //Add components to the main JFrame
        contentPane.add(boardPanel);
        contentPane.add(buttonPanel);

        board.pack();
        board.setResizable(false);
        board.setVisible(true);

    }

    /**
     *
     * @return NO_OF_COLOMNS
     */
    public int getNoOfCols(){
        return noOfColomns;
    }

    /**
     *
     * @return NO_OF_ROWS
     */
    public int getNoOfRows(){
        return noOfRows;
    }
```

```
    /**
     *
     * @return BOARD_SIZE
     */
    public Dimension getBoardSize(){
        return boardSize;
    }

    public void update(java.util.Observable o, Object arg){
        if(gameModel.boardIsEmpty()){
            if(reset.isEnabled()){
                reset.setEnabled(false);
                endGame.setEnabled(false);
            }
        }else{
            if(reset.isEnabled()==false){
                reset.setEnabled(true);
                endGame.setEnabled(true);
            }
        }
        board.repaint();

    }

    /**
     * Called by the colomnPanel class when mouse is clicked on a colomn. Passes
     * click details to gameController.mouseClickedOnPiece()
     *
     * @param y         The Y coordinate of the click
     * @param colomnNo  The number of the colomnPanel that called
     *                  the method and hence the colomn.
     */
    public void mouseClicked(int y, int colomnNo){
        gameController.mouseClickedOnPiece(y, colomnNo);

    }


    /**
     * Displays a dialogue when a player wins
     * @param player    Player number of the winning player
     */
    public void winningPlayerDialog(int player){
        JOptionPane.showMessageDialog(board, "Player "+player+" wins!");
    }

    public void actionPerformed(ActionEvent event){
        if(event.getSource()==endGame){
            gameController.endGame();
        }
        if(event.getSource()==reset){
            gameController.resetScores();
        }
        if(event.getSource()==cpu){
            gameController.setCPU();

            if(gameController.getCPU()){
```

```
                cpu.setText("De-activate CPU");
            }else{
                cpu.setText("Activate CPU");
            }

        }

    }

}
```