```java
/*
 * 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 javax.swing.Timer;
import java.util.Observer;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

/**
 *
 * @author User
 */
public class FPController {

    private FPModel gameModel;
    private FPView gameView;
    private CPUPlayer cpuOpponent;
    int startingPlayer,winningPlayer,currentPlayer,cpuPlayer;
    boolean cpuActivated;


    public FPController(FPModel gameModel){
        startingPlayer = 1;
        winningPlayer = 0;
        currentPlayer = 1;
        cpuPlayer = 1;
        cpuActivated = false;

        this.gameModel = gameModel;
        cpuOpponent = new CPUPlayer(gameModel,this,cpuPlayer);

    }

    public void setView(FPView gameView){
        this.gameView = gameView;
    }
    /**
     * This method processes a move from a human player and will be bypassed
     * if the current player is the CPU. The CPU only toggles the player variable
     * once it has completed a move and hence locks this method out during a CPU move.
     *
     * @param y          Y coordinate clicked
     * @param colomnNo   Colomn clicked on
     */
    public void mouseClickedOnPiece(int y, int colomnNo){

        if((cpuActivated && currentPlayer != cpuPlayer)||(!cpuActivated)){
            if(winningPlayer==0){
                int colomnWidth,row,rowHeight;
                int[][] boardStatus = gameModel.getChipStatus();
                colomnWidth = gameView.getBoardSize().width/gameView.getNoOfCols();
                rowHeight = gameView.getBoardSize().height/gameView.getNoOfRows();
                row = (gameView.getNoOfRows()-1)-((int)y/rowHeight);

                //Checks if the move is valid and if it is performs is.
                if(gameModel.validMove(row, colomnNo)){
                    gameModel.setPiece(colomnNo, currentPlayer);
                    togglePlayer();
                    checkForWinner();

                    //Set by checkForWinner()
                    if(winningPlayer == 0){
                        if((cpuActivated) &&(currentPlayer==cpuPlayer)){
                            cpuOpponent.doMove();
                        }
                    }
                }
            }
        }
    }
    /**
     * Called by FPView when the end game button is pressed. This
     * method clears the board and swaps the starting player.
     */
    public void endGame(){
        gameModel.clearBoard();
        if(winningPlayer > 0){
            winningPlayer = 0;
        }
        if(startingPlayer ==  1)
            startingPlayer = 2;
        else
            startingPlayer = 1;
        currentPlayer=startingPlayer;
        if(cpuActivated){
            if(currentPlayer==cpuPlayer){
                cpuOpponent.doMove();
            }
        }
    }

    /**
     * Called by FPView to reset the scores when the reset button
     * is pressed.
     */
    public void resetScores(){
        gameModel.setScore(1, 0);
        gameModel.setScore(2, 0);
    }

    /**
     * Called by the FPView to activate the CPU Player
     */
    public void setCPU(){
        if(cpuActivated){
            cpuActivated =  false;
        }else{
            cpuActivated  =  true;
            if(currentPlayer == 1){
                cpuOpponent.doMove();
            }
        }
    }

    /**
     * @return  cpuActivated
     */
    public boolean getCPU(){
        return cpuActivated;
    }

    /**
     * This method toggles the current player
     */
    public void togglePlayer(){
        if(currentPlayer == 1){
            currentPlayer = 2;
        }else{
            currentPlayer = 1;
        }
    }

    /**
     * This method checks for a winner using the method in the model winningLine()
     * and sets the winningPlayer variable accordingly. Also activates a dialogue.
     */
    public void checkForWinner(){
        if(gameModel.winningLine(1)){
            winningPlayer = 1;
            gameView.winningPlayerDialog(winningPlayer);
            gameModel.setScore(winningPlayer, gameModel.getScore(winningPlayer)+1);
        }
        if(gameModel.winningLine(2)){
            winningPlayer = 2;
            gameView.winningPlayerDialog(winningPlayer);
            gameModel.setScore(winningPlayer, gameModel.getScore(winningPlayer)+1);
        }

    }

}
```