```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.*;
import java.awt.*;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;

/**
 *
 * @author Lee Hudson 09092543
 * This class is a specialised JPanel that displays a single colomn of
 * the board. It handles all clicking events associated with it.
 *
 */
public class ColomnPanel extends JPanel implements MouseListener {

    private Dimension size;
    private int colomnNumber;
    private int chipWidth,chipHeight,verticalSpacing,x,y,noOfChips;
    FPModel model;
    FPView gameView;


    public ColomnPanel(int width, int height, int noOfChips, int colomnNumber, FPModel model,
    FPView gameView){

        this.size = new Dimension(width,height);
        chipWidth=(int)(width-(width*0.1));
        chipHeight=(int)(height/noOfChips-((height/noOfChips)*0.1));
        x=(size.width-chipWidth)/2;
        this.noOfChips=noOfChips;
        y=(height/noOfChips)-chipHeight;
        this.colomnNumber=colomnNumber;
        this.model = model;
        this.gameView=gameView;
        this.addMouseListener(this);
    }


    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        int[][] boardStatus = model.getChipStatus();
        for(int i=0; i < noOfChips; i++){
            switch(boardStatus[(noOfChips-1)-i][colomnNumber]){
                case 0:
                    g.setColor(Color.GRAY);
                    break;
                case 1:
                    g.setColor(Color.RED);
                    break;
                case 2:
                    g.setColor(Color.BLACK);
                    break;
                default:
                    g.setColor(Color.GRAY);
                    break;
            }
            g.fillOval(x,((chipHeight+y)*i),chipWidth,chipHeight);
        }
    }

    /**
     *
     * @return colomnNumber
     */
    public int getColomnNumber(){
        return colomnNumber;
    }

    @Override
    public Dimension getPreferredSize(){
        return size;
    }

    @Override
    public Dimension getMinimumSize(){
        return getPreferredSize();
    }

    @Override
    public Dimension getMaximumSize(){
        return getPreferredSize();
    }


    public void mouseClicked(MouseEvent e) {
    }

    public void mousePressed(MouseEvent e) {
        gameView.mouseClicked(e.getY(), colomnNumber);
    }
    public void mouseReleased(MouseEvent e) {
    }
    public void mouseEntered(MouseEvent e) {
    }
    public void mouseExited(MouseEvent e) {
    }

    public void mouseMoved(MouseEvent e) {

    }

    public void mouseDragged(MouseEvent e) {
    }

}
```