```java
import java.awt.*;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.util.ArrayList;

import javax.swing.*;

public class BoardPanel extends JPanel implements KeyListener {
	private Player player;
	private ArrayList<Monster> monsters;
	private Grid grid;
	private final int cellWidth = 35;
	private final int cellHeight = 35;
	private final int Lmargin = 100;
	private final int Tmargin = 40;

	public BoardPanel(Grid grid, Player player, ArrayList<Monster> monsters) {
		this.player = player;
		this.grid = grid;
		this.monsters = monsters;

	}

	// reset game
	public void reset(Grid grid, Player player, ArrayList<Monster> monsters) {
		this.player = player;
		this.grid = grid;
		this.monsters = monsters;
	}

	/* responds to various Keyboard pressed */
	@Override
	public void keyPressed(KeyEvent ke) {
		if (ke.getKeyCode() == KeyEvent.VK_LEFT) {
			if (player.getDirection() != 'L') {
				player.clearPress();
			}
			player.setDirection('L');
			player.addPress();
		}
		if (ke.getKeyCode() == KeyEvent.VK_RIGHT) {
			if (player.getDirection() != 'R') {
				player.clearPress();
			}
			player.setDirection('R');
			player.addPress();
		}
		if (ke.getKeyCode() == KeyEvent.VK_UP) {
			if (player.getDirection() != 'U') {
				player.clearPress();
			}
			player.setDirection('U');
			player.addPress();
		}
		if (ke.getKeyCode() == KeyEvent.VK_DOWN) {
			if (player.getDirection() != 'D') {
				player.clearPress();
			}
			player.setDirection('D');
			player.addPress();
		}
		if (ke.getKeyCode() == KeyEvent.VK_Z) {
			player.putTrap();
		}
		if (ke.getKeyCode() == KeyEvent.VK_X) {
			player.putBlock();
		}
	}

	@Override
	public void keyReleased(KeyEvent ke) {
	}

	@Override
	public void keyTyped(KeyEvent e) {
	}

	/* returns the x coordinate based on left margin and cell width */
	private int xCor(int col) {
		return Lmargin + col * cellWidth;
	}

	/* returns the y coordinate based on top margin and cell height */
	private int yCor(int row) {
		return Tmargin + row * cellHeight;
	}

	/*
```

```
         * Redraws the board and the pieces Called initially and in response to
         * repaint()
         */
        protected void paintComponent(Graphics graphics) {
                super.paintComponent(graphics);
                Cell cells[] = grid.getAllCells();
                Cell cell;
                for (int i = 0; i < cells.length; i++) {
                        cell = cells[i];
                        if (cell.col % 5 == 0 && cell.row % 5 == 0)
                                graphics.setColor(Color.cyan);
                        else
                                graphics.setColor(Color.white);
                        graphics.fillRect(xCor(cell.col), yCor(cell.row), cellWidth, cellHeight);
                        graphics.setColor(Color.black);
                        graphics.drawRect(xCor(cell.col), yCor(cell.row), cellWidth, cellHeight);
                        if (cell.gotGold) {
                                graphics.setColor(Color.MAGENTA);
                                graphics.fillArc(xCor(cell.col) + cellWidth / 8, yCor(cell.row) +
cellHeight / 8, cellWidth * 3 / 4,
                                                cellHeight * 3 / 4, 45, 45);
                                graphics.setColor(Color.white);
                                graphics.drawString("G", xCor(cell.col) + cellWidth / 3, yCor(cell.row) + 2
* cellWidth / 3);
                        }
                }
                cell = player.getCell();
                graphics.setColor(Color.red);
                graphics.fillOval(xCor(cell.col) + cellWidth / 8, yCor(cell.row) + cellHeight / 8, cellWidth
* 3 / 4,
                                cellHeight * 3 / 4);
                graphics.setColor(Color.white);
                graphics.drawString("P", xCor(cell.col) + cellWidth / 3, yCor(cell.row) + 2 * cellWidth /
3);

                for (Trap trap : player.getTrap()) {
                        if (trap.getState()) {
                                cell = trap.getCell();
                                graphics.setColor(Color.green);
                                graphics.fillRect(xCor(cell.col), yCor(cell.row), cellWidth, cellHeight);
                                graphics.setColor(Color.white);
                                graphics.drawString("T", xCor(cell.col) + cellWidth / 3, yCor(cell.row) + 2
* cellWidth / 3);
                        }
                }

                for (Roadblock roadblock : player.getBlock()) {
                        if (roadblock.getState()) {
                                cell = roadblock.getCell();
                                graphics.setColor(Color.blue);
                                graphics.fillRect(xCor(cell.col), yCor(cell.row), cellWidth, cellHeight);
                                graphics.setColor(Color.white);
                                graphics.drawString("B", xCor(cell.col) + cellWidth / 3, yCor(cell.row) + 2
* cellWidth / 3);
                        }
                }
                for (Monster monster : monsters) {
                        cell = monster.getCell();
                        if (monster.viewable() && !monster.isBaby()) {
                                graphics.setColor(Color.black);
                                graphics.fill3DRect(xCor(cell.col) + cellWidth / 8, yCor(cell.row) +
cellHeight / 8, cellWidth * 3 / 4,
                                                cellHeight * 3 / 4, true);
                                graphics.setColor(Color.white);
                                graphics.drawString("M", xCor(cell.col) + cellWidth / 3, yCor(cell.row) + 2
* cellWidth / 3);
                        } else if (monster.viewable() && monster.isBaby()) {
                                graphics.setColor(Color.yellow);
                                graphics.fill3DRect(xCor(cell.col) + cellWidth / 8, yCor(cell.row) +
cellHeight / 8, cellWidth * 3 / 4,
                                                cellHeight * 3 / 4, true);
                                graphics.setColor(Color.white);
                                graphics.drawString("B", xCor(cell.col) + cellWidth / 3, yCor(cell.row) + 2
* cellWidth / 3);
                        }
                }
        }
}
```