

```
package hu.elte.prt.eightqueens.view;

import java.awt.Color;
import java.awt.Component;
import java.awt.Dimension;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.JOptionPane;

import hu.elte.prt.eightqueens.model.Engine;

public class Frame extends JFrame {

    private static final long serialVersionUID = 8316572961171616624L;

    private Engine engine;
    private JMenuItem pauseMenuItem;

    public Frame(Engine engine) {
        super("8 Queens");
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        this.engine = engine;
    }

    public void showFrame() {
        createFields();
        setMenu();
        pack();
        setVisible(true);
    }

    private void createFields() {
        getContentPane().setLayout(new GridLayout(engine.getSize(), engine.getSize()));
        for (int i = 0; i < engine.getSize(); ++i) {
            for (int j = 0; j < engine.getSize(); ++j) {
                JButton field = new JButton();
                field.setBackground(getFieldBackground(i, j));
                field.setPreferredSize(new Dimension(80, 80));
                field.setFont(field.getFont().deriveFont(30.0f));
                addFieldActionListener(i, j, field);
                getContentPane().add(field);
            }
        }
    }

    private void addFieldActionListener(int i, int j, JButton field) {
        field.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent arg0) {
                engine.put(i, j);
            }
        });
    }
}
```

```
        updateFields();
        checkVictoryCondition();
    }
});
}

private void checkVictoryCondition() {
if (engine.won()) {
    JOptionPane.showMessageDialog(this, "Oh nice.");
    updateFields();
}
}

private void setMenu() {
JMenuBar menuBar = new JMenuBar();
JMenu menu = new JMenu("Game");
menuBar.add(menu);
addRestartMenuItem(menu);
addUndoMenuItem(menu);
addPauseMenuItem(menu);
setJMenuBar(menuBar);
}

private void addRestartMenuItem(JMenu menu) {
JMenuItem menuItem = new JMenuItem("Restart");
menuItem.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent arg0) {
        engine.startNewGame();
        updateFields();
        updatePauseMenuItem();
    }
});
menu.add(menuItem);
}

private void addUndoMenuItem(JMenu menu) {
JMenuItem menuItem = new JMenuItem("Undo");
menuItem.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent arg0) {
        engine.undo();
        updateFields();
    }
});
menu.add(menuItem);
}

private void addPauseMenuItem(JMenu menu) {
pauseMenuItem = new JMenuItem("Pause");
pauseMenuItem.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent arg0) {
        engine.togglePause();
        updatePauseMenuItem();
        updateFields();
    }
}
```

```
});  
menu.add(pauseMenuItem);  
}  
  
private void updatePauseMenuItem() {  
    pauseMenuItem.setText(engine.isPaused() ? "Resume" : "Pause");  
}  
  
private void updateFields() {  
    for (int i = 0; i < engine.getSize(); ++i) {  
        for (int j = 0; j < engine.getSize(); ++j) {  
            Component c = getContentPane().getComponent(i * engine.getSize() + j);  
            JButton field = (JButton) c;  
            field.setText(getFieldText(i, j));  
            field.setBackground(getFieldBackground(i, j));  
        }  
    }  
}  
  
private String getFieldText(int i, int j) {  
    return !engine.isPaused() && engine.isQueen(i, j) ? "♔" : "";  
}  
  
private Color getFieldBackground(int i, int j) {  
    if (!engine.isPaused() && engine.canPutHere(i, j)) {  
        return Color.RED;  
    }  
    return (i + j) % 2 == 0 ? Color.GRAY : Color.WHITE;  
}  
}
```