Fck code

2017. január 13.

Fck

```java
package game;

import game.model.Engine;
import game.view.Frame;

public class Fasz {

    public static void main(String[] args) {
        Engine engine = new Engine();
        engine.startNewGame();
        Frame frame = new Frame(engine);
        frame.showFrame();

    }

}
```

Frame

```java

package game.view;

import game.model.Engine;
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;

@SuppressWarnings("ALL")
public class Frame extends JFrame{

    private Engine engine;

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

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

    private void setMenu() {
        JMenuBar menuBar = new JMenuBar();
        JMenu menu = new JMenu("Game");
        menuBar.add(menu);

        addRestartMenuItem(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();

            }
        });
        menu.add(menuItem);
    }

    private void createFields() {
        getContentPane().setLayout(new
GridLayout(Engine.SIZE,Engine.SIZE));
        for(int i=0;i < Engine.SIZE;++i){
            for(int j=0;j < Engine.SIZE;++j){
                JButton button = new JButton();
                button.setPreferredSize(new Dimension(80, 80));
                addActionListener(button, i,j);
                getContentPane().add(button);
            }
        }

    }

    private void addActionListener(JButton button, int i, int j) {
        button.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                engine.put(i, j);
                updateFields();
                checkVictoryCondition();
            }
        });
    }

    public void updateFields(){
        for (int i = 0; i < Engine.SIZE; ++i) {
            for (int j = 0; j < Engine.SIZE; ++j) {
                Component c = getContentPane().getComponent(i * Engine.SIZE
+ j);
```

```
                int red = 100;
                int green = 210;
                int blue = 200;
                Color col = new Color(red, green, blue);
                Color col2 = new Color(255, 229, 204);
                JButton field = (JButton) c;
                field.setText(Integer.toString(engine.get(i, j)));
                field.setBackground(engine.get(i, j) % 2 == 0 ? col :
col2);
            }
        }
    }

    private void checkVictoryCondition() {
        if (engine.isGameOver()) {
            JOptionPane.showMessageDialog(this, "TE egy kibaszott zseni
vagy.");
            updateFields();
        }
    }
}
```

Engine

```
package game.model;

import java.util.Random;

public class Engine {

    public static final int SIZE = 7;

    private int[][] field;

    public void startNewGame(){
        field = new int[SIZE][SIZE];
        for(int i =0; i < SIZE;++i){
            for(int j =0; j < SIZE;++j){
                field[i][j] = new Random().nextInt(52);
            }
        }
    }

    public void put(int i, int j){
        for(int k =0;k < SIZE;++k){
            if(k != j){
                field[i][k]--;
            }
        }
        for(int l =i;l >=0;--l){
            if(l != i){
                field[l][j]--;
            }
        }
    }

    public int get(int i, int j){
        return field[i][j];
    }

    public boolean isGameOver(){
        for(int j =0;j < SIZE;++j){
            for(int i =0;i < SIZE-1;++i){
                if(field[i][j]< field[i+1][j]){
                    return false;
                }
            }
            try{
                if(field[SIZE-1][j]< field[0][j+1]){
                    return false;
                }
            }catch(ArrayIndexOutOfBoundsException e){

            }

        }
        return true;
    }

}
```