

# Mastermind

2017. január 16.

## Palette (enum)

```
package game;

import java.awt.*;

public enum Palette {

    WHITE(Color.WHITE), BLACK(Color.BLACK), RED(255, 51, 51), GREEN(51, 204, 51), BLUE(
    51, 153, 255), YELLOW(255, 255, 102);

    Palette(int r, int g, int b) {
        this.color = new Color(r, g, b);
    }

    public Color getColor() {
        return color;
    }

    private final Color color;

    Palette(Color color) {
        this.color = color;
    }
}
```

## Launcher

```
package game;

import game.model.MastermindEngine;
import game.view.MastermindFrame;

public class Launcher {
    public static void main(String[] args) {
        MastermindEngine engine = new MastermindEngine();
        engine.startNewGame();
        MastermindFrame frame = new MastermindFrame(engine);
        frame.showFrame();
    }
}
```

# Mastermind

2017. január 16.

## Engine

```
package game.model;

import game.Palette;
import java.awt.*;
import java.util.*;
import java.util.List;

public class MastermindEngine {

    public static final int TRY_SIZE = 10;
    public static final int COLOR_COUNTY = 4;
    private static final Palette[] colours = Palette.values();
    private Palette[][] userschoice;
    private Palette[] field;
    private int actualRow;

    public void startNewGame() {
        actualRow = 0;
        field = new Palette[COLOR_COUNTY];
        userschoice = new Palette[TRY_SIZE][COLOR_COUNTY];
        for (int i = 0; i < COLOR_COUNTY; i++) {
            Palette p = getRandomColor();
            field[i] = p;
            System.out.println(p.name());
        }
    }

    private Palette getRandomColor() {
        return colours[new Random().nextInt(colours.length)];
    }

    public boolean isGameOver() {
        return actualRow > TRY_SIZE;
    }

    public boolean isGameOverWin() {
        return actualRow > 0 && actualRow <= TRY_SIZE &&
isCorrect(actualRow - 1) == 4 ;
    }

    public void click(int i, int j) {
        if (actualRow == i) {
            if (userschoice[i][j] == null) {
                userschoice[i][j] = colours[0];
            }
            int index = 0;
            for (int k = 0; k < colours.length; k++) {
                if (colours[k] == userschoice[i][j]) {
                    index = k;
                    break;
                }
            }
            if (index == colours.length - 1) {
                index = 0;
            } else {

```

# Mastermind

2017. január 16.

```
        index++;
    }
    userschoice[i][j] = colours[index];
}

public Color getBackgroundColor(int i, int j) {
    if (userschoice[i][j] == null) {
        if(actualRow == i){
            return null;
        }
        return Color.GRAY;
    } else {
        return userschoice[i][j].getColor();
    }
}

public int isCorrect(int j) {
    int counter = 0;
    for(int i = 0; i < COLOR_COUNTY; i++) {
        if (userschoice[j][i] != null && userschoice[j][i] ==
(field[i])) {
            counter++;
        }
    }
    return counter;
}

public int isInside(int i) {
    Set<Palette> choices = new
HashSet<>(Arrays.asList(userschoice[i]));
    int counter = 0;
    List<Palette> fields = Arrays.asList(field);
    for(Palette p : choices) {
        if (fields.contains(p) ) {
            counter++;
        }
    }
    return counter;
}

public void next() {
    ArrayList<Palette> choices = new
ArrayList<>(Arrays.asList(userschoice[actualRow]));
    choices.removeIf(i -> i == null);
    if(choices.size() == COLOR_COUNTY){
        actualRow++;
    }
}
}
```

# Mastermind

2017. január 16.

## Frame

```
package game.view;

import game.model.MastermindEngine;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyEvent;
import javax.swing.*;

import static game.model.MastermindEngine.COLOR_COUNTY;
import static game.model.MastermindEngine.TRY_SIZE;

public class MastermindFrame extends JFrame {

    public static final int BWIDTH = 80;
    public static final int BHEIGHT = 60;
    private MastermindEngine engine;
    private JPanel gamepanel;
    private JPanel labelpanel;
    private JPanel buttonpanel;

    public MastermindFrame(MastermindEngine engine) {
        super("Mastermind Simple");
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        gamepanel = new JPanel();
        gamepanel.setLayout(new GridLayout(TRY_SIZE, COLOR_COUNTY));
        getContentPane().add(gamepanel, BorderLayout.WEST);
        labelpanel = new JPanel();
        labelpanel.setLayout(new GridLayout(TRY_SIZE, COLOR_COUNTY));
        getContentPane().add(labelpanel, BorderLayout.EAST);
        buttonpanel = new JPanel();
        buttonpanel.setLayout(new FlowLayout(FlowLayout.CENTER));
        getContentPane().add(buttonpanel, BorderLayout.SOUTH);
        this.engine = engine;
    }

    public void showFrame() {

        createFields();
        createLabels();
        setBottomButton();
        pack();
        setMenu();
        setVisible(true);
    }

    private void createLabels() {
        for (int i = 0; i < TRY_SIZE; i++) {
            labelpanel.add(createLabel());
        }
    }

    private JLabel createLabel() {
```

# Mastermind

2017. január 16.

```
JLabel label = new JLabel("Szerepelt : 0 ** Eltalált : 0");
return label;
}

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

private void setBottomButton() {
    JButton bottombutton = new JButton("Next");
    bottombutton.setPreferredSize(new Dimension(80, 60));
    bottombutton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            engine.next();
            updateLabels();
            checkVictoryCondition();
            updateFields();
        }
    });
    buttonpanel.add(bottombutton);
}

private void updateLabels() {
    for(int i = 0 ; i < TRY_SIZE;i++){
        Component c = labelpanel.getComponent(i);
        JLabel label = (JLabel) c;
        updateLabel(label,i);
    }
}

private void updateLabel(JLabel label,int i) {
    int qty = engine.isInside(i);
    int inPlace = engine.isCorrect(i);
    label.setText("Szerepelt : "+qty+" ** Eltalált : "+ inPlace);
}

private void addRestartMenuItem(JMenu menu) {
    JMenuItem menuItem = new JMenuItem("New game");
    menuItem.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent arg0) {
            engine.startNewGame();
            updateFields();
            updateLabels();
        }
    });
    menuItem.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_F1, 0));
    menu.add(menuItem);
}

private void createFields() {
```

# Mastermind

2017. január 16.

```
        for (int i = 0; i < TRY_SIZE; i++) {
            for (int j = 0; j < COLOR_COUNTY; j++) {
                JButton button = new JButton();
                button.setPreferredSize(new Dimension(BWIDTH, BHEIGHT));
                addActionListener(button, i, j);
                gamepanel.add(button);
            }
        }

        updateFields();
    }

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

    public void updateFields() {
        for (int i = 0; i < TRY_SIZE; i++) {
            for (int j = 0; j < COLOR_COUNTY; j++) {
                Component c = gamepanel.getComponent(i * COLOR_COUNTY + j);
                JButton field = (JButton) c;
                field.setBackground(engine.getBackgroundColor(i, j));
            }
        }
    }

    private void checkVictoryCondition() {
        if (engine.isGameOverWin()) {
            int answer = JOptionPane.showConfirmDialog(this,
                "Congratulations Bitch.Wanna play again?" , "GameOver" ,
                JOptionPane.YES_NO_OPTION);
            if (answer == JOptionPane.YES_OPTION) {
                engine.startNewGame();
                updateFields();
                updateLabels();
            } else {
                System.exit(0);
            }
        }
        if (engine.isGameOver()) {
            JOptionPane.showMessageDialog(this, "Fail.");
        }
    }
}
```