

2048

DV-Projekt



Dominik Butscher, Robert Voss, Moritz Schneider,
Adrian Martin, Eric Miller, Nick Rothmund



Gliederung

1. Spielerklärung
2. Planungsphase / Zielsetzung
3. Umsetzung
4. Erklärung/ Vorführung GUI
5. Erklärung Spiel
6. Erklärung Highscore
7. Fragen

2048

Play

Highscore

Settings

Quit

2048

SCORE**62**

Spielerklärung

- 4x4 Feld mit zwei Startzahlen
- Verschieben der Zahlen mit den Pfeiltasten
- Ziel ist die Bildung der Zahl "2048"
- Game Over wenn kein Zug mehr möglich

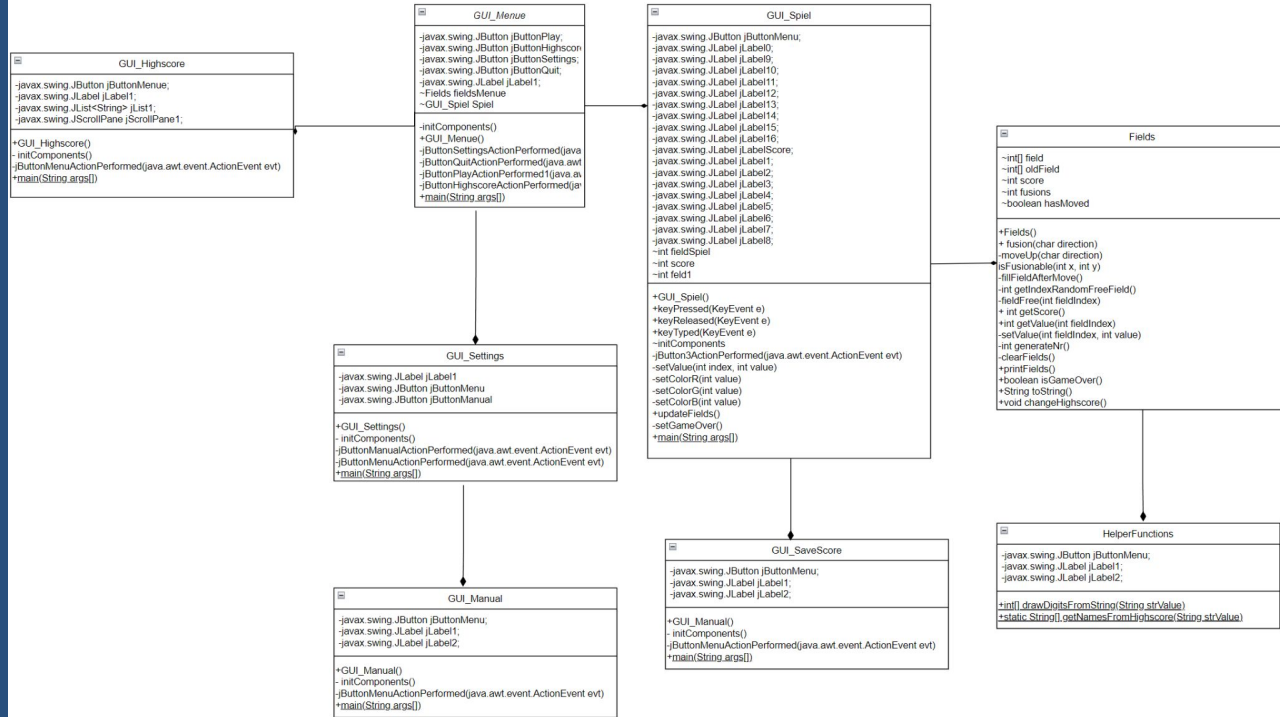
2	2	32	2
	4		16
			2
	2		

Planungsphase / Zielsetzung

- Einteilung abklären
- Ziel war es
Hauptanforderungen sowie
nicht-funktionale
Anforderungen zu erfüllen



Umsetzung



Erklärung/ Vorführung GUI

- Siehe Programmfenster

```
4 public class GUI_Menue extends javax.swing.JFrame {
5
6     private javax.swing.JButton jButtonPlay;
7     private javax.swing.JButton jButtonHighscore;
8     private javax.swing.JButton jButtonSettings;
9     private javax.swing.JButton jButtonQuit;
10    private javax.swing.JLabel jLabel1;
11
12    Fields fieldsMenue;
13    GUI_Spiel Spiel;
14
15    public GUI_Menue() {
16        System.out.println("GUI_Menue");
17        initComponents();
18    }
19
20    private void initComponents() {
21
22        jLabel1 = new javax.swing.JLabel(); // 2048
23        // jLabel2 = new javax.swing.JLabel();
24        jButtonPlay = new javax.swing.JButton(); // Spiel starten
25        jButtonHighscore = new javax.swing.JButton(); // Highscoreliste öffnen
26        jButtonSettings = new javax.swing.JButton(); // Einstellungen öffnen
27        jButtonQuit = new javax.swing.JButton(); // Anwendung schließen
28
29        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
30        setTitle("2048 Menue");
31        setLocation(new java.awt.Point(0, 0));
32        // setLocation(new java.awt.Point(x));
33        setName("2048");
34        setPreferredSize(new java.awt.Dimension(400, 500));
35        setResizable(false);
36        setType(java.awt.Window.Type.POPUP);
37
38        // Point x = getLocationOnScreen();
39
40        jLabel1.setFont(new java.awt.Font("Arial Rounded MT Bold", 0, 80));
41        jLabel1.setForeground(new java.awt.Color(0, 136, 82));
42        jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
43        jLabel1.setText("2048");
44    }
45 }
```



Erklärung Spiel

Erklärung Highscore

Erster versuch mit java und einem
Web-server

website konnte aber nur design und
keine Anfragen bearbeiten.

zweiter Versuch mit Github.

```
import java.io.*;
import java.net.*;

public class ServerConnection{
    {
        try {
            URL url = new URL("https://dv-projekt-2022-2048.jimdofree.com");
            HttpURLConnection conn = (HttpURLConnection) url.openConnection();
            conn.setRequestMethod("POST");
            conn.setRequestProperty("Content-Type", "application/json");
            DataOutputStream writer = new DataOutputStream(conn.getOutputStream());
            writer.write(1);
            writer.flush();
            writer.close();
            wait(500);
            conn.setRequestMethod("GET");
            BufferedReader input = new BufferedReader(new InputStreamReader(conn.getInputStream()));
            String line;
            StringBuilder content = new StringBuilder();
            while((line = input.readLine()) != null)
            {
                content.append(line);
                content.append(System.lineSeparator());
            }
            conn.disconnect();
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
}
```


Erklärung Highscore

Python download

Über ein Github
access-Token wird auf die .md
Datei zugegriffen

```
from msilib.schema import File
from github import Github
import os

g = Github("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")
repo = g.get_repo("RoBeatHfu/DV-Projekt")
directory = os.path.dirname(os.path.abspath(__file__))
filename = os.path.join(directory, "Highscore.md")
file = repo.get_contents("Highscore.md")
highscore = open(filename, "w")
content = str(file.decoded_content)
content = content.lstrip(content[0])
content = content.strip("'")
print(content)
highscore.write(content)
```

Erklärung Highscore

Python upload

```
from msilib.schema import File
from github import Github
import os

g = Github("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")      #connect to
repo = g.get_repo("RoBeatHfu/DV-Projekt")                   #get our pro
directory = os.path.dirname(os.path.abspath(__file__))       #get path to
filename = os.path.join(directory, "Highscore.md")          #make it an
file = repo.get_contents("Highscore.md")                    #get content
highscoreFile = open(filename, "r")
highscoreList = highscoreFile.readlines()
highscoreFile.close()
highscore = ""
for player in highscoreList:
    highscore = highscore + player
print(highscore)
repo.update_file(file.path, "update of Highscore", highscore, file.sha)
```

Erklärung Highscore

Highscore nach Spiel

Highscore Runterladen
(python starten)

String erstellen

```
public void changeHighscore() {  
    if (isGameOver()) {  
        String everything;  
        int achievedScore = getScore();  
        String path = System.getProperty("user.dir");  
        String command = "python " + path + "\\GitHubDownload.py";  
        try {  
            Process p = Runtime.getRuntime().exec(command);  
            BufferedReader br = new BufferedReader(new FileReader("Highscore.md"));  
            try {  
                StringBuilder sb = new StringBuilder();  
                String line = br.readLine();  
  
                while (line != null) {  
                    sb.append(line);  
                    sb.append(System.lineSeparator());  
                    line = br.readLine();  
                }  
                everything = sb.toString();  
            } finally {  
                br.close();  
            }  
        }  
    }  
}
```

Highscore Erklärung

Highscore in Zahlen und
Buchstaben aufteilen

dabei überflüssige
Leerzeichen Trimmen

```
public class HelperFunctions {  
  
    public static int[] drawDigitsFromString(String strValue){  
        String str = strValue;  
        String digits="";  
        int highscore[] = {0,0,0};  
        int index = 0;  
        for (int i = 0; i < str.length(); i++) {  
            char chrs = str.charAt(i);  
            if(chrs == '\n') {  
                if(digits != "")  
                {  
                    digits.trim();  
                    highscore[index] = Integer.parseInt(digits);  
                    index++;  
                    digits = "";  
                }  
            }  
            if (Character.isDigit(chrs))  
                digits = digits+chrs;  
        }  
        return highscore;  
    }  
}
```

Highscore Erklärung

.md Datei in Zahlen und
Buchstaben aufteilen

Da top 3, String array mit 3
plätzen

fehler mit Absatz vor Namen
behoben.

```
public static String[] getNamesFromHighscore(String strValue){
    String str = strValue;
    String letters="";
    String names[] = {"", "", ""};
    int index = 0;
    for (int i = 0; i < str.length(); i++) {
        char chrs = str.charAt(i);
        if(chrs == '\n') {
            names[index] = letters;
            index++;
            letters = "";
        }
        if (!Character.isDigit(chrs))
            if(chrs != '\r' && chrs != '\n')
                letters = letters+chrs;
        }
    return names;
}
```

Highscore Erklärung

Sortierung der top 3

helper variablen um punktzahl
und Platzierung einzufügen.

wenn 1. oder 2. Platz erreicht,
werden vorherige verschoben.

wenn 3. Platz erreicht werden
vorherige Überschrieben.

```
int actualHighscore[] = HelperFunctions.drawDigitsFromString(everything);
String[] names = HelperFunctions.getNamesFromHighscore(everything);
for(int i = 0; i < 3; i++) {
    names[i].trim();
    int helper;
    String helperName;
    if(achievedScore > actualHighscore[i]) {
        helper = actualHighscore[i];
        actualHighscore[i] = achievedScore;
        helperName = names[i];
        names[i] = "player";
        if(i == 0) {
            int helper2 = actualHighscore[1];
            actualHighscore[1] = helper;
            actualHighscore[2] = helper2;
            String helperName2 = names[1];
            names[1] = helperName;
            names[2] = helperName2;
            break;
        }
        if(i == 1) {
            actualHighscore[2] = helper;
            names[2] = helperName;
            break;
        }
    }
}
```


Highscore Erklärung

Zusammenführen und
Hochladen

neuer String wird erstellt und
dann mit python hochgeladen

```
1      String output = names[0] + " " + actualHighscore[0] + "\n" + names[1] + " " +  
5          actualHighscore[1] + "\n" + names[2] + actualHighscore[2];  
5      PrintWriter out = new PrintWriter("Highscore.md");  
7      out.write(output);  
8      out.close();  
9      command = "python " + path + "\\GitHubUpload.py";  
9      Process p2= Runtime.getRuntime().exec(command);  
1     } catch (IOException e) {  
2         // TODO Auto-generated catch block  
3         e.printStackTrace();  
4     }  
5 }  
5 }  
7 }
```

**Vielen Dank für
Ihre Aufmerksamkeit**

