

SPŠE Ječná

Informační technologie

Ječná 30, 120 00, Praha 2

KARETNÍ KASINO

Ondřej Fejl

Informační technologie

2023

Obsah

| | |
|-------------------------------|----------|
| <u>Cíl Práce</u> | 1 |
| <u>Software</u> | 1 |
| <u>Program</u> | 1 |
| <u>Závěr</u> | 1 |
| <u>Zdroje</u> | 1 |

1 Cíl práce

Cílem mého závěrečného projektu bylo vytvoření podobné sázející hry jako v mobilní aplikaci **RallyAces Poker**. Dalším mým cílem bylo zasáhnout do GUI, a tedy naučit se pracovat s grafikou v Java.

2 Software

Celý můj projekt je psán v jazyce Java ve verzi 1.8.0_51 v prostředí IntelliJ IDEA ve verzi 2022.3.

3 Program

Celá hra nejprve spustí úvodní obrazovku s přivítáním a dvěma tlačítky s pravidly a samotnou hrou. O komponenty na úvodní stránku a stránku s pravidly se starají třídy **WelcomeIntro** a **RulesComponents**. V nich se vytvářejí panely, labely a tlačítka, které se následně vloží do hlavních tříd.

Tlačítkem PLAY! se spustí okno, ve kterém celá hra probíhá. Jako první, co je nutné, je vsazení částky, o což se stará třída **Betting**. Dále zvolení možnosti, na co vsadit, právě o to se starají třídy **Options** a **OptionButtons**. Ve třídě **Options** je asi nejvíce proměnných z celého projektu, a to proto, že do jednoho labelu se mi nepodařilo vložit více jak jeden řádek, a tak jsem byl nucen vytvořit pro každý řádek textu vlastní label a zároveň i panel.

```
10 public class Options {
11     JPanel Spades;
12     JPanel SpadesChance;
13     JPanel Hearts;
14     JPanel HeartsChance;
15     JPanel Clubs;
16     JPanel ClubsChance;
17     JPanel Diamonds;
18     JPanel DiamondsChance;
19     JPanel Low;
20     JPanel LowChance;
21     JPanel High;
22     JPanel HighChance;
23     JPanel Red;
24     JPanel RedChance;
25     JPanel Black;
26     JPanel BlackChance;
27 }
```

```
29 public Options() {
30
31     //region SPADES
32     JLabel SpadesLabel = new JLabel();
33     SpadesLabel.setText("ONLY SPADES");
34     SpadesLabel.setVerticalAlignment(JLabel.BOTTOM);
35     SpadesLabel.setHorizontalAlignment(JLabel.CENTER);
36     SpadesLabel.setForeground(Color.orange);
37     SpadesLabel.setFont(new Font("Calibri", Font.ITALIC, 24));
38
39     JLabel SpadesLabelChance = new JLabel();
40     SpadesLabelChance.setText("1:2.06");
41     SpadesLabelChance.setVerticalAlignment(JLabel.BOTTOM);
42     SpadesLabelChance.setHorizontalAlignment(JLabel.CENTER);
43     SpadesLabelChance.setForeground(Color.orange);
44     SpadesLabelChance.setFont(new Font("Calibri", Font.ITALIC, 24));
45
46     Spades = new JPanel();
47     Spades.setBounds(0, 0, 170, 200, 200, 35);
48     Spades.setBackground(new Color(1, 0.8, 0.28));
49     Spades.setLayout(new BorderLayout());
50     Spades.add(SpadesLabel);
51
52     SpadesChance = new JPanel();
53     SpadesChance.setBounds(0, 0, 205, 200, 200, 35);
54     SpadesChance.setBackground(new Color(1, 0.8, 0.28));
55     SpadesChance.setLayout(new BorderLayout());
56     SpadesChance.add(SpadesLabelChance);
57
58     //endregion
59     //region HEARTS
60     JLabel HeartsLabel = new JLabel();
61     HeartsLabel.setText("ONLY HEARTS");
```

Zároveň také obsahuje v konstruktoru samotné naplnění panelů a labelů, což je vloženo do samostatných regionů. Třída **OptionButtons** je asi nejrozsáhlejší, neboť v ní se generují karty, vytvářejí se tlačítka s volbou a vytváří se výherní okno.

Za zmínku také stojí třídy Card a DeckOfCards, kde ve třídě **Card** je vše připraveno pro vytvoření karty a ve třídě **DeckOfCards** se všechny karty generují a zapisují do kolekce. Tato třída také obsahuje metodu na míchání a vybrání tří karet, které se zobrazují na hlavním hracím poli a také metodu pro zjištění, zda uživatel vyhrál nebo ne, a to pomocí dvou vstupních proměnných, které jsou tři vylosované karty a možnost, kterou si uživatel vybral.

```
public boolean AnalyzeWin(String Option, ArrayList<Card> ThreeCards) {  
  
    boolean WinFlag = false;  
  
    switch (Option) {  
        case "spades" -> {  
            if (ThreeCards.get(0).getTypeCard().equals(ThreeCards.get(1).getTypeCard()) && ThreeCards.get(1).getTypeCard().equals("spades")) {  
                WinFlag = true;  
            } else if (ThreeCards.get(0).getTypeCard().equals(ThreeCards.get(2).getTypeCard()) && ThreeCards.get(0).getTypeCard().equals("spades")) {  
                WinFlag = true;  
            } else if (ThreeCards.get(1).getTypeCard().equals(ThreeCards.get(2).getTypeCard()) && ThreeCards.get(2).getTypeCard().equals("spades")) {  
                WinFlag = true;  
            }  
        }  
        case "hearts" -> {  
            if (ThreeCards.get(0).getTypeCard().equals(ThreeCards.get(1).getTypeCard()) && ThreeCards.get(1).getTypeCard().equals("hearts")) {  
                WinFlag = true;  
            } else if (ThreeCards.get(0).getTypeCard().equals(ThreeCards.get(2).getTypeCard()) && ThreeCards.get(0).getTypeCard().equals("hearts")) {  
                WinFlag = true;  
            } else if (ThreeCards.get(1).getTypeCard().equals(ThreeCards.get(2).getTypeCard()) && ThreeCards.get(2).getTypeCard().equals("hearts")) {  
                WinFlag = true;  
            }  
        }  
        case "clubs" -> {  
            if (ThreeCards.get(0).getTypeCard().equals(ThreeCards.get(1).getTypeCard()) && ThreeCards.get(1).getTypeCard().equals("clubs")) {  
                WinFlag = true;  
            } else if (ThreeCards.get(0).getTypeCard().equals(ThreeCards.get(2).getTypeCard()) && ThreeCards.get(0).getTypeCard().equals("clubs")) {  
                WinFlag = true;  
            } else if (ThreeCards.get(1).getTypeCard().equals(ThreeCards.get(2).getTypeCard()) && ThreeCards.get(2).getTypeCard().equals("clubs")) {  
                WinFlag = true;  
            }  
        }  
        case "diamonds" -> {  
            if (ThreeCards.get(0).getTypeCard().equals(ThreeCards.get(1).getTypeCard()) && ThreeCards.get(1).getTypeCard().equals("diamonds")) {  
                WinFlag = true;  
            } else if (ThreeCards.get(0).getTypeCard().equals(ThreeCards.get(2).getTypeCard()) && ThreeCards.get(0).getTypeCard().equals("diamonds")) {  
                WinFlag = true;  
            } else if (ThreeCards.get(1).getTypeCard().equals(ThreeCards.get(2).getTypeCard()) && ThreeCards.get(2).getTypeCard().equals("diamonds")) {  
                WinFlag = true;  
            }  
        }  
    }  
    return WinFlag;  
}
```

4 Závěr

Ačkoli jsem na začátku nevěděl, zda vůbec dosáhnu nějakého výsledku práce, jelikož jsem s grafikou ještě nepracoval, jsem rád za výsledek, který vznikl. Myslím, že má hra by mohla zaujmout všechny, kteří rádi zkoušejí nové karetní hry nebo rádi sází bez rizika ztracených peněz.

5 Zdroje

[Java GUI: Full Course !\[\]\(8af806fb1314382d09bc5ec5b767526c_img.jpg\) \(FREE\) - YouTube](#)

[Google Code Archive - Long-term storage for Google Code Project Hosting.](#)

<https://stackoverflow.com/questions/9864267/loading-resources-like-images-while-running-project-distributed-as-jar-archive>

<https://stackoverflow.com/questions/21056815/where-should-i-put-my-resources-for-a-java-program>