

Gymnázium, Praha 6, Arabská 14
Obor programování



Ročníkový projekt 1.ročník

Filip Beneš

Ruleta

Duben 2021

Prohlašuji, že jsem jediným autorem tohoto projektu, všechny citace jsou řádně označené a všechna použitá literatura a další zdroje jsou v práci uvedené. Tímto dle zákona 121/2000 Sb. (tzv. Autorský zákon) ve znění pozdějších předpisů uděluji bezúplatně škole Gymnázium, Praha 6, Arabská 14 oprávnění k výkonu práva na rozmnožování díla (§ 13) a práva na sdělování díla veřejnosti (§ 18) na dobu časově neomezenou a bez omezení územního rozsahu.

V Praze dne 02.05.2021

Filip Beneš

Název práce: Ruleta

Autor: Filip Beneš

Anotace:

Cílem projektu bylo navrhnout virtuální podobu hry ruleta. Program má své vlastní tokeny, tudíž není potřeba vkládat reálné peníze. Hra není vytvořena pro výdělek. Program je napsán v programovacím jazyce Java. Program má své vlastní grafické rozhraní, které je napsáno v Java Swing a lze jej jednoduše ovládat díky několika málo tlačítkům. Program umožňuje sázet nejen na jedno číslo, ale i na barvu a skupinu čísel.

Klíčová slova: Tokeny, Sázka na číslo, Sázka na barvu, Sázka na skupinu čísel

Obsah

1. Úvod	5
1.1. Zadání	5
1.2. Pravidla	6
2. Obsah kódu.....	7
2.1. Náhodné číslo.....	7
2.2. Animace točení Rulety.....	7
2.3. Vyhodnocování sázky na barvu	8
2.4. Vyhodnocování sázky na skupinu čísel.....	8
2.5. Vypisování výhry prohry.....	9
3. Chyby a podmínky	10
3.1. Podmínky před roztočením rulety.....	10
3.2. Podmínky při vypisování výhry, prohry	10
3.3. Hra skončila	11
4. Grafické rozhraní	12
4.1. Prostředí hry.....	12
4.2. Základní okno	12
4.3. Vedlejší okna	13
5. Instalace a spuštění	14
6. Závěr	14
7. Seznam obrázků	15
8. Bibliografie	15

1. Úvod

V této dokumentaci o ročníkovém projektu se zabývám koncepcí a realizací hry pro lidi, kteří mají rádi hazardní hry. Inspirací pro projekt mi byly stolní hry, které jsou na stejném principu jako tato. Vždy když jsme hráli s kamarády obdobné hry, prodělával jsem herní žetony. To mě přinutilo k nápadu vytvořit hazardní hru, která bude nastavená co možná nejvíce „férově“ (hráč neprodělá vždy celý vklad).

Projekt není vytvořený pro vkládání reálných peněz a poté jejich sázení. Je pouze pro zábavné hraní a vyzkoušení si této hry. Ve hře se nachází virtuální tokeny, které jsou obdobou herních žetonů.

1.1. Zadání

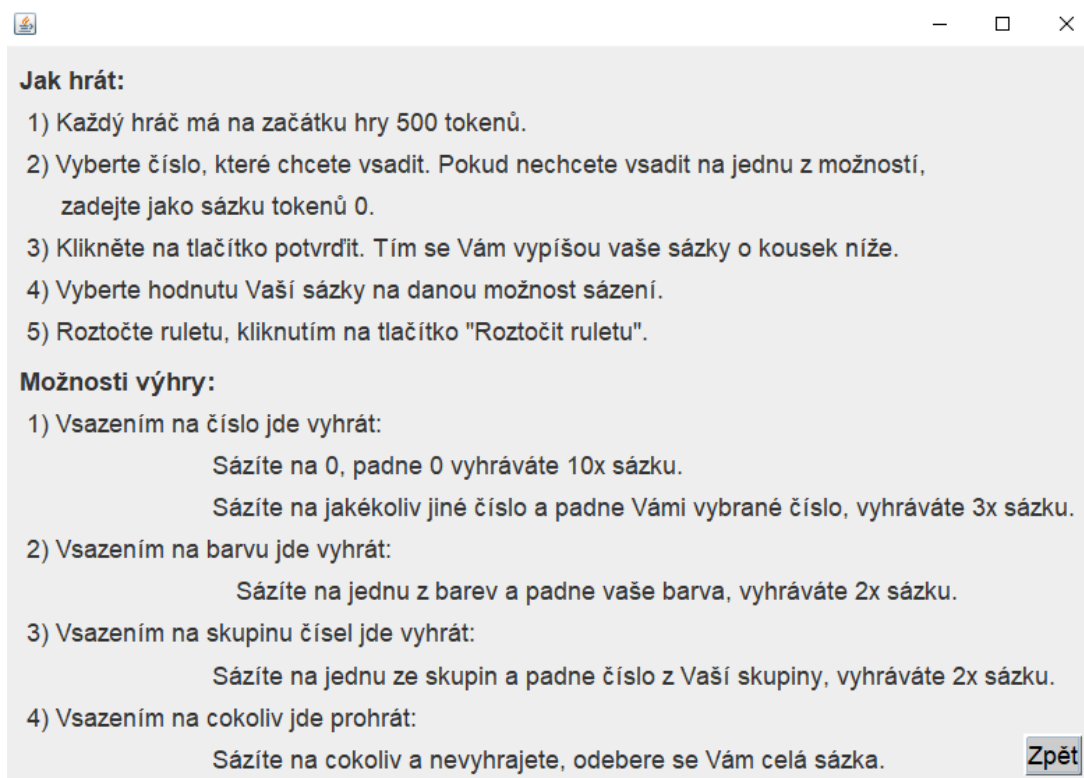
V ruletě budou čísla od 0-36 a zároveň tři barvy (zelená = 0, černá = všechna lichá čísla, červená = všechna sudá čísla mimo 0). Jako první na začátku budete mít 500 tokenů. Bude možné vsadit na jedno číslo, jednu barvu nebo skupinu čísel. Poté přijde na řadu, kolik budete chtít z vkladu vsadit na určitou možnost. Následně se "roztočí" ruleta a vybere jedno náhodné číslo, pokud je stejné jako to, na které sázíte, program by měl přidat do vkladu trojnásobek vsazení. Když padne Vámi vsazená barva vyhráváte dvojnásobek sázky. Dvojnásobek se Vám přičte i pokud padne Vámi vybraná skupina čísel. Jestliže nenastane ani jedna z předchozích možností ztrácíte celou Vaši sázku. Největší zisk by měl jít získat při vsazení čísla 0. Když by se povedlo vylosovat 0 a vy jste na ni vsadili dostanete desetinásobek vaší sázky.

Na konec programu by se mělo vypsát:

1. Aktuální zisk (ztráta)
2. Počet točení
3. Chcete hrát znovu? Ano\Ne

1.2. Pravidla

Pravidla celé hry jsou sepsána v jednom vytvořeném dialogu. Lze je otevřít kliknutím na stejnojmenné tlačítko. V pravidlech je pomocí několika bodů vysvětlený postup hry. Dále jsou v pravidlech vysvětleny všechny možné výhry a prohry. Do dialogu je přidáno tlačítko, které už podle názvu napovídá, že se vrátíte na hlavní okno.



Obrázek 1-1 Pravidla hry

2. Obsah kódu

V této kapitole je popsán obsah kódu u programu. Jsou zde popsány jen ty více důležité a zajímavé části kódu.

2.1. Náhodné číslo

Náhodné číslo je losováno jednoduše pomocí metody. Metoda se nachází ve třídě Math v Java Docs a nemusí se ani importovat. Potřeboval jsem, aby metoda losovala jenom čísla od 0 do 36. Tento problém jsem vyřešil násobením random metody číslem 37. Musel jsem to násobit 37, neboť náhodné číslo je desetinné a mohlo by nastat, že násobením 36 se číslo 36 vylosuje jen s velmi malou pravděpodobností. Abych zamezil tomu, že se vylosuje číslo 37 a abych z desetinných čísel vytvořil celá čísla, obalil jsem metodu random metodou floor, opět ve třídě Math.

```
NahodneCislo = (int) (Math.floor(Math.random() * 37));
```

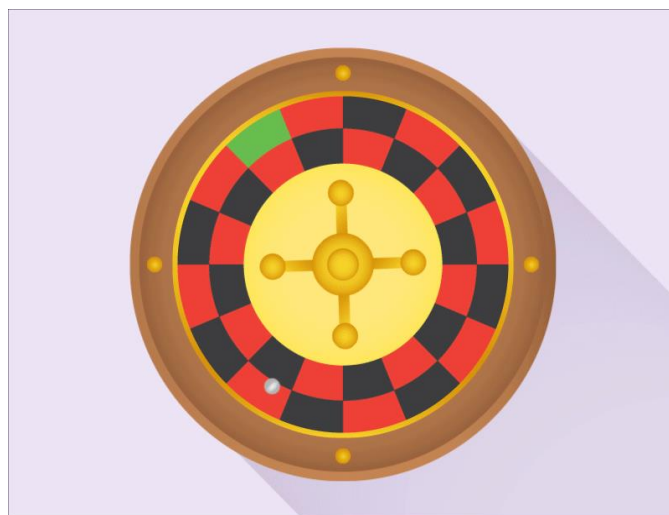
Obrázek 2-1 Generátor náhodného čísla

2.2. Animace točení Rulety

Animace točení rulety je vytvořená z klasického obrázku, který se po kliknutí vymění, ale na stejný pohybující se GIF, který jsem si vypůjčil na internetu. (1) Pro výměnu obrázků je využita metoda setIcon, (2) která je volně dostupná v třídě Java. K vytvoření časovače jsem použil třídu Timer a pro vytvoření zadání k časovači jsem využil třídu TimerTask (3). Třídě Timer jsem zadal metodu schedule a zadal jsem jí můj TimerTask a zpoždění (4). Aby hráč nemohl mačkat tlačítko pro roztočení rulety pořád dokola, tak se po kliknutí tlačítko zneviditelná na dobu, kdy je zapnutý časovač. Animace je ošetřená výjimkou InterruptedException (5), která zabráňuje chybám během přerušení programu.

```
TimerTask task = new TimerTask() {
    public void run() {
        try {
            RoztocitKoloJButton.setEnabled(false);
            jLabel13.setIcon(new javax.swing.ImageIcon(getClass().getResource("/IMG/casino-roulette.gif")));
            TimeUnit.SECONDS.sleep(3);
        } catch (InterruptedException ex) {
            System.out.println("Chyba animace rulety");
        }
        jLabel13.setIcon(new javax.swing.ImageIcon(getClass().getResource("/IMG/casino-roulette02.jpg")));
        RoztocitKoloJButton.setEnabled(true);
    }
};
dobaToceni = (int) (Math.floor((Math.random() * 100) + dobaToceni));
Timer timer = new Timer();
long delay = dobaToceni;
timer.schedule(task, delay);
```

Obrázek 2-2 Animace točení rulety



Obrázek 2-3 Ruleta

2.3. Vyhodnocování sázky na barvu

V programu existují 2 barvy červená a černá. Červená obsahuje všechny sudá čísla v Ruletě mimo 0, jelikož 0 je řazena do zelené barvy. Na zelenou barvu není potřeba sázet a hráči stačí vsadit pouze na číslo 0 = „zelená barva“. Černá naopak obsahuje všechny lichá čísla v Ruletě. Nejdříve se pomocí podmínky if vybere barva, na kterou bylo vsazeno. Pomocí for cyklu jsem poté vytvořil skupinu čísel od 1 do 36. Pro vytvoření sudých nebo lichých skupin jsem použil podmínku if a to tak, že buď po dělení 2 vyjde zbytek 0 = sudé, nebo zbytek 1 = liché číslo.

```
if (barva == "Červená") {
    for (int cervena = 2; cervena <= 36; cervena++) { // pro červenou barvu (sudá čísla)
        if (cervena % 2 == 0 && cervena == NahodneCislo && barva == "Červená") {
            ZiskBarva = SazkaTokenBarva * 1;
        } else if (NahodneCislo % 2 == 1 || NahodneCislo == 0) { // Když prohrajete
            ZiskBarva = SazkaTokenBarva * -1;
        }
    }
} else if (barva == "Černá") {
    for (int cerna = 1; cerna <= 36; cerna++) { // pro černou barvu (lichá čísla)
        if (cerna % 2 == 1 && cerna == NahodneCislo && barva == "Černá") {
            ZiskBarva = SazkaTokenBarva * 1;
        } else if (NahodneCislo % 2 == 0 || NahodneCislo == 0) { // Když prohrajete
            ZiskBarva = SazkaTokenBarva * -1;
        }
    }
}
```

Obrázek 2-4 Vyhodnocování sázky na barvu

2.4. Vyhodnocování sázky na skupinu čísel

Vyhodnocování sázky na skupinu čísel je velmi podobné vyhodnocení sázky na barvu. Nejdříve jsem pomocí podmínek if vybral skupinu, na kterou bylo vsazeno. Pomocí for cyklu jsem vytvořil danou skupinu čísel. Poté jsem pomocí podmínky if vybral, jestli je číslo stejné

jako vylosované, pokud bylo, hráč získal dvojnásobek sázky. Pokud ne hráč ztratil vše, co vsadil a program pokračuje dál.

```
if (skupinaCisel == "1-12") {
    for (int skupCis = 1; skupCis <= 12; skupCis++) {
        if (skupCis == NahodneCislo && skupinaCisel == "1-12") {
            ZiskSkupinaCisel = SazkaTokenSkupinaCisel * 1;
        } else if (NahodneCislo > 12 || skupinaCisel != "1-12" || NahodneCislo == 0) {
            ZiskSkupinaCisel = SazkaTokenSkupinaCisel * -1;
        }
    }
} else if (skupinaCisel == "13-24") {
    for (int skupCis = 13; skupCis <= 24; skupCis++) {
        if (skupCis == NahodneCislo && skupinaCisel == "13-24") {
            ZiskSkupinaCisel = SazkaTokenSkupinaCisel * 1;
        } else if (NahodneCislo > 24 || NahodneCislo < 13 || skupinaCisel != "13-24" || NahodneCislo == 0) {
            ZiskSkupinaCisel = SazkaTokenSkupinaCisel * -1;
        }
    }
} else if (skupinaCisel == "25-36") {
    for (int skupCis = 25; skupCis <= 36; skupCis++) {
        if (skupCis == NahodneCislo && skupinaCisel == "25-36") {
            ZiskSkupinaCisel = SazkaTokenSkupinaCisel * 1;
        } else if (NahodneCislo < 25 || skupinaCisel != "25-36" || NahodneCislo == 0) {
            ZiskSkupinaCisel = SazkaTokenSkupinaCisel * -1;
        }
    }
}
```

Obrázek 2-5 Výhodnocování sázky na skupinu čísel

2.5. Vypisování výhry prohry

Po skončení animace se vypíše výhra, prohra, aktuální zisk, celkový zisk a počet točení. Aktuální Zisk, ať už záporný nebo nezáporný, se získá pomocí sečtení zisků ze všech tří kategorií sázení. Zisk celkový, v programu nazvaný jako Ziskpo, se získá sečtením zisku aktuálního a celkového. Po každém roztočení se změní i stav tokenů, který se sečte z aktuálních tokenů a aktuálního zisku. Jako poslední program ještě vypíše počet točení rulety. Ten se počítá způsobem, že po každém roztočení se přičte do proměnné jedna a vypíše se.

```
// vypisování prohry nebo výhry
Zisk = Ziskbarva + ZiskCislo + ZiskSkupinaCisel;
Ziskpo = Zisk + Ziskpo;
tokeny = tokeny + Zisk;
JedenZiskjLabel.setText(String.valueOf(Zisk));
ZiskCislojLabel.setText(String.valueOf(Ziskpo));
PocetTokenVvojLabel1.setText(String.valueOf(tokeny));

// Kolikrát jste už točili
pocetToceni++;
PocetToceniLabel.setText(String.valueOf(pocetToceni));
```

Obrázek 2-6 Vypisování výhry, prohry a počet točení

3. Chyby a podmínky

Všechny chyby, které by mohly nastat jsem ošetřoval pomocí podmínek if. Výjimka Try a catch je v programu využita jenom v jednom případě. A to v případě, kdy bylo ošetření chyb pomocí podmínky if náročnější. Výjimku jsem použil u části kódu, kde se zabývám animací rulety.

3.1. Podmínky před roztočením rulety

Před roztočením kola jsem musel vytvořit několik podmínek pomocí if.

- Jedna podmínka měla za úkol nepustit hráče k roztočení kola, dokud nepotvrdí změnu čísla sázky.
- Druhá má obdobný úkol, jenom platí na skupinu čísel, třetí platí na sázení barvy.
- Čtvrtá má za úkol ohlídat hráče, jestli nesází více tokenů než kolik jich vlastní.
- Pátá hlídá situaci, kdy hráč vsadí na vše nula. V takovém případě ho hra upozorní, že se kolo nemůže roztočit, dokud nevsadí alespoň na jednu z možností jeden token.
- Existuje ještě jedna podmínka před roztočením kola, a to že hráč musí vybrat na jakou možnost bude sázet, pokud tak neučiní, program mu vypíše hlášku: “Musíte vybrat, na co sázíte.”

```
if (CslSa != potvrdCislo) {  
    PotvrdCislojLabel.setText(String.valueOf("Nejdříve potvrdte číslo"));  
} else if ((SazkaTokenCislo + SazkaTokenBarva + SazkaTokenSkupinaCisel) > tokeny) {  
    PotvrdCislojLabel.setText(String.valueOf("Sázíte moc velké číslo"));  
} else if (barva != String.valueOf(SazkaComboBox.getSelectedItem())) {  
    PotvrdCislojLabel.setText(String.valueOf("Nejdříve potvrdte barvu"));  
} else if (skupinaCisel != JakaSkupinaCiselSazkaLabel18.getText()) {  
    PotvrdCislojLabel.setText(String.valueOf("Nejdříve potvrdte skupinu čísel"));  
} else if ((SazkaTokenCislo + SazkaTokenBarva + SazkaTokenSkupinaCisel) == 0) {  
    PotvrdCislojLabel.setText(String.valueOf("Sázíte na vše 0"));  
} else {  
    PotvrdCislojLabel.setText(String.valueOf("Musíte vybrat na co sázíte"));  
}
```

Obrázek 3-1 Podmínky před roztočením rulety

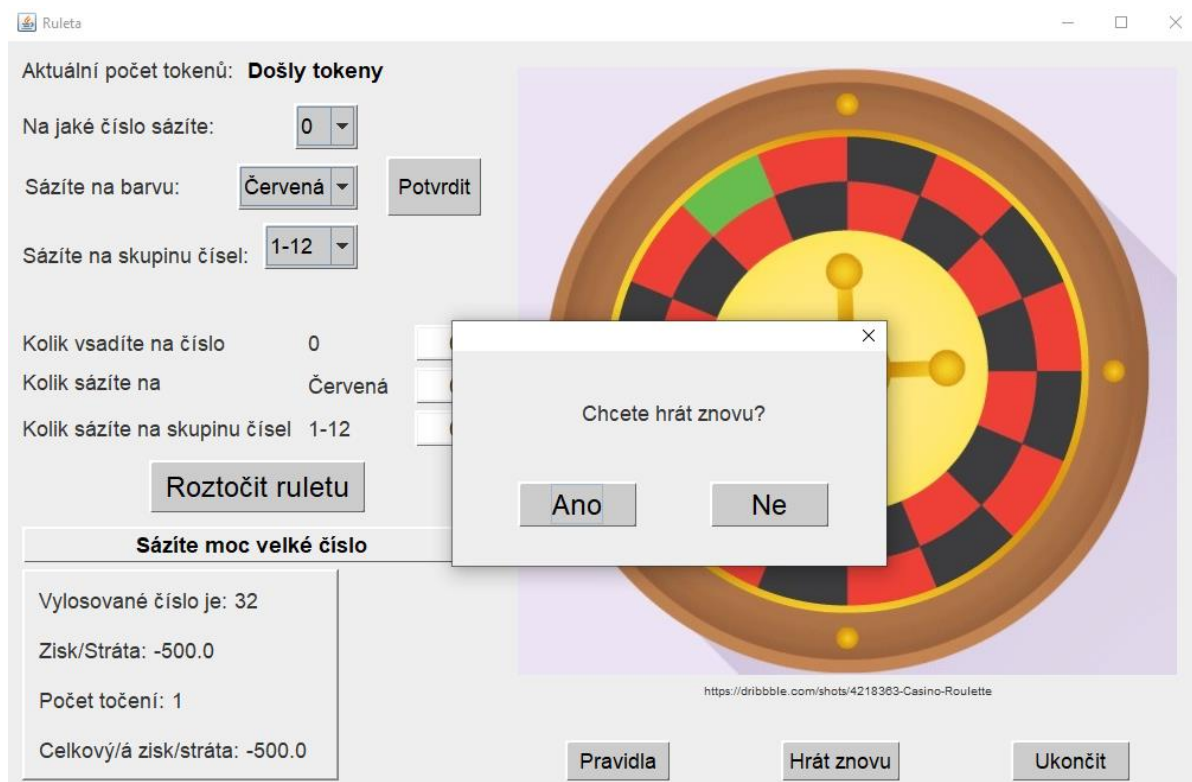
3.2. Podmínky při vypisování výhry, prohry

Po roztočení kola a skončení animace program musí vypsát výhru nebo prohru a další statistiky, pro vypsaní jsou vytvořené následující podmínky:

- Vypiš výhru, prohru, pokud hráč vlastní ještě nějaké tokeny.
- Vypiš výhru, pokud nedošlo k porušení žádné z výše uvedených podmínek (kapitola 3.1.).

3.3. Hra skončila

Když program zjistí pomocí podmínky if, že hráč vlastní nula nebo méně tokenů, nepovolí hráči točit ruletou. Otevře se dialog, který se zeptá hráče, jestli chce hrát znovu nebo jestli chce opustit hru. Po kliknutí na tlačítko ANO, tedy hrát znovu, se hra vynuluje, v kódu metoda ZnovuHrat, a vše vypadá jako po prvním spuštění hry. Po kliknutí na tlačítko NE, tedy nechci už znovu hrát, se celý program vypne. (6)



Obrázek 3-2 takto vypadá program poté, co hráči dojdou tokeny

4. Grafické rozhraní

4.1. Prostředí hry

Grafické prostředí je vytvořené v Java Swing. (5) Hra se ovládá tlačítky vytvořenými v grafickém návrháři GUI (Graphical User Interface), který se stáhne spolu s aplikací NetbeansIDE, ve které jsem projekt vytvořil. Po spuštění programu se otevře hlavní grafické okno, na kterém probíhá celá hra. Všechny tlačítka jsou popsána a celá hra by měla být ošetřená proti podvodníkům nebo proti nechtěnému kliknutí na tlačítko.

4.2. Základní okno

Nejvíce místa v základním okně zabírá velký obrázek rulety. Po kliknutí na odkaz se z obrázku stane GIF (1), který má představovat animaci roztočené rulety.

Vlevo nahoře se nachází velmi důležité číslo. Toto číslo udává, kolik v tuto chvíli vlastníte tokenů. Každý hráč na začátku hry dostane vždy 500 tokenů. Po každém roztočení kola se počet tokenů aktualizuje na aktuální hodnotu.

Pod počtem tokenů se nachází tři comboBoxi, díky nimž si hráč vybere z nabízených možností, na co přesně chce sázet. Vpravo od comboboxů je tlačítko, na které hráč musí kliknout pokaždé, kdy změní jednu ze sázek.

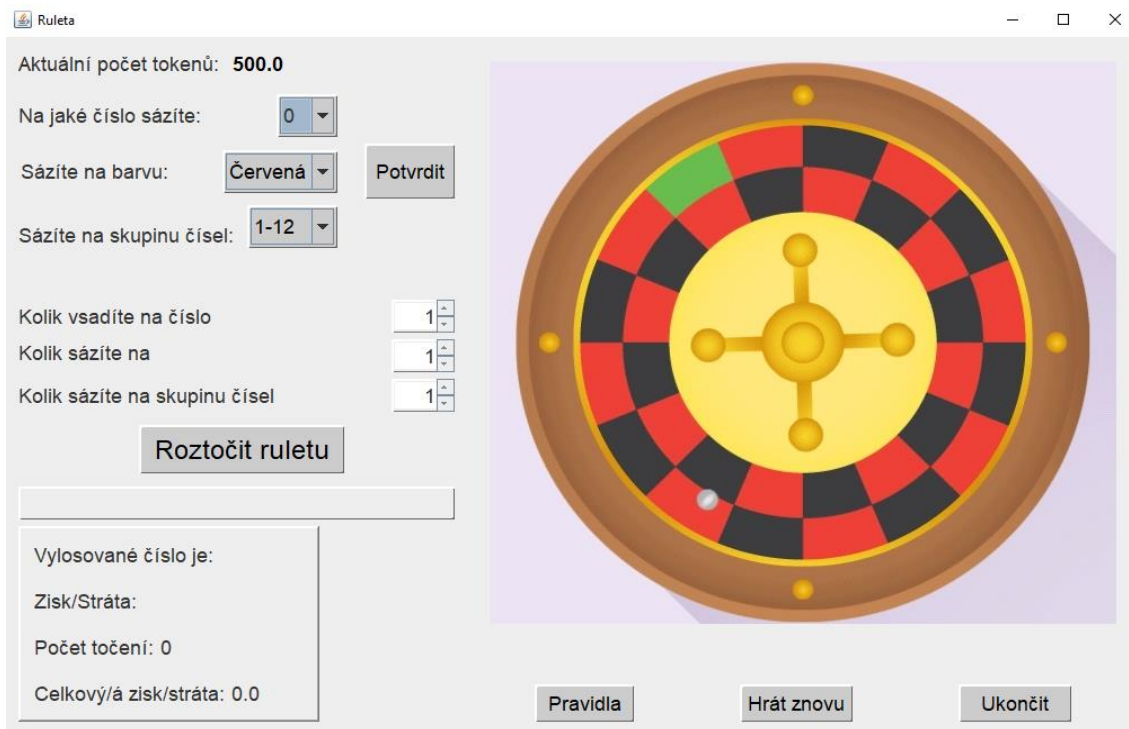
O kousek níže jsou tři spinnery, jež udávají počet tokenů, které chcete vsadit na danou možnost.

Vlevo dole je ohraničený obdélník, do kterého se po každém roztočení rulety vypíší výhry, prohry atd...

Nad tímto obdélníkem je ohraničený pruh, do něhož se vypisují všechny nastávající chybné fráze.

Přibližně vlevo uprostřed je vytvořené tlačítko, které po kliknutí spustí celou hru, tzn. "roztočí ruletu."

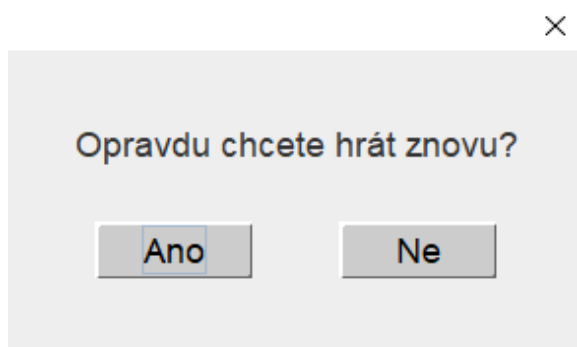
V hlavním okně jsou ještě tři menší tlačítka, která zastávají funkci, podle toho, co je na nich napsáno.



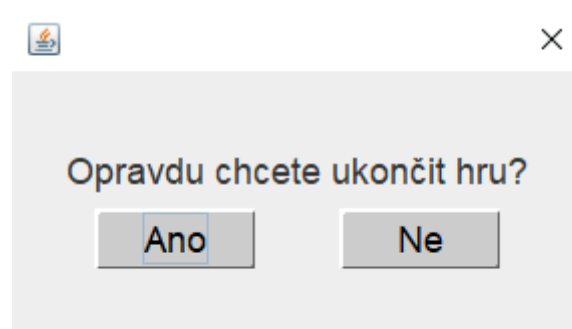
Obrázek 4-1 Hlavní okno programu

4.3. Vedlejší okna

V této hře se kromě hlavního okna nacházejí ještě tři vedlejší dialogy, které se otevřou po kliknutí na určité tlačítko. Dva z nich mají velmi podobnou funkci a je možné na ně odpovědět kliknutím na tlačítko „Ano“ nebo „Ne“. První je pro uzavření hry, druhý pro vynulování hry (hrát znovu). Velmi důležitý je poslední dialog, který po kliknutí na tlačítko otevře pravidla hry.



Obrázek 4-2 Vedlejší okno pro hraní znovu



Obrázek 4-3 Vedlejší okno pro ukončení hraní

5. Instalace a spuštění

Instalovat a spustit program lze jednoduše. Ke stažení souboru s programem můžete použít tento odkaz na [GitHub](#). Po stažení souboru se musíte ujistit, jestli máte v počítači staženou příslušnou verzi Javy. Pokud se ve vašem počítači nenachází potřebná verze Javy, nepůjde otevřít. Jestliže se ve Vašem počítači Java nachází, stačí dvojklikem kliknout na soubor a tím se Vám otevře okno programu. Jako první se doporučuji podívat na pravidla hry, a to pomocí kliknutí na tlačítko pravidla.

6. Závěr

Závěrem tento projekt hodnotím úspěšně. Svoji úroveň programování jsem díky němu posunul o pár úrovní výše. Přesto jsem musel překonat některé překážky, a to hlavně při sepisování dokumentu. Nicméně nakonec vše dopadlo nad má očekávání.

Když jsem si projekt vybíral a sepisoval zadání, mým cílem bylo vytvořit hru, která bude, pokud možno férová, jelikož tomu tak u hazardních her často není. Zkoušel jsem vše, co bylo v mých silách, ale dospěl jsem k závěru, že hazardní hry jsou ve výsledku stejně pouze o náhodě.

V projektu jsem používal grafické rozhraní Java Swing, což teď hodnotím jako ne příliš dobrou volbu. V případě vytváření nového projektu bych určitě využil spíše Java FX z důvodu popularity u jiných programátorů.

Nakonec bych rád přidal své doporučení, které spočívá ve zdržení se dlouhodobého hraní této i obdobných hazardních her. Vždy po dlouhodobém hraní existuje minimum lidí, kteří mají štěstí a tím pádem i zisk.

7. Seznam obrázků

Obrázek 1-1 Pravidla hry	6
Obrázek 2-1 Generátor náhodného čísla	7
Obrázek 2-2 Animace točení rulety	7
Obrázek 2-3 Ruleta.....	8
Obrázek 2-4 Vyhodnocování sázky na barvu	8
Obrázek 2-5 Vyhodnocování sázky na skupinu čísel	9
Obrázek 2-6 Vypisování výhry, prohry a počet točení	9
Obrázek 3-1 Podmínky před roztočením rulety	10
Obrázek 3-2 takto vypadá program poté, co hráči dojdou tokeny	11
Obrázek 4-1 Hlavní okno programu.....	13
Obrázek 4-3 Vedlejší okno pro hraní znovu	13
Obrázek 4-2 Vedlejší okno pro ukončení hraní	13

8. Bibliografie

1. **Zadeh, Islam.** dribbble.com. *Casino Roulette*. [Online] [Citace: 20. březen 2021.]
<https://dribbble.com/shots/4218363-Casino-Roulette>.
2. **Maadawi, Amro Read El.** stackoverflow.com. *How to use the geticon function?* [Online] [Citace: 5. duben 2021.] <https://stackoverflow.com/questions/9457659/how-to-use-the-geticon-function-java-net-beans-ide>.
3. **Baeldung.** baeldung.com. *Java - Timer | Baeldung*. [Online] [Citace: 24. březen 2021.]
<https://www.baeldung.com/java-timer-and-timertask>.
4. **příspěvatelé, GeeksforGeeks a.** geeksforgeeks.org. *TimeUnit Class in Java with Examples*. [Online] 02. únor 2021. [Citace: 26. březen 2021.]
<https://www.geeksforgeeks.org/timeunit-class-in-java-with-examples/>.
5. **Java, dokumentace.** oracle.com. *Třída InterruptedException*. [Online] [Citace: 14. duben 2021.] <https://docs.oracle.com/javase/7/docs/api/java/lang/InterruptedException.html>.
6. **Community, Stackoverflow.** stackoverflow.com. *Terminating a Java Program*. [Online] 23. květen 2017. [Citace: 25. březen 2021.]
<https://stackoverflow.com/questions/22452930/terminating-a-java-program>.

7. **ITNetwork.** ITNetwork.cz. *Základy Java Swing*. [Online] [Citace: 30. březen 2021.]

<https://www.itnetwork.cz/java/swing/zaklady>.

8. **W3Schools.** w3schools.com. *Java Exceptions (Try...Catch)*. [Online] [Citace: 27. březen

2021.] https://www.w3schools.com/java/java_try_catch.asp.