

Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

Hra Replacement puzzle



2019

Vedoucí práce: doc. RNDr. Mi-
roslav Kolařík, Ph.D.

Filip Tůma

Studijní obor: Informatika, prezenční
forma

Bibliografické údaje

Autor: Filip Tůma
Název práce: Hra Replacement puzzle
Typ práce: bakalářská práce
Pracoviště: Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby: 2019
Studijní obor: Informatika, prezenční forma
Vedoucí práce: doc. RNDr. Miroslav Kolařík, Ph.D.
Počet stran: 31
Přílohy: 1 CD
Jazyk práce: český

Bibliographic info

Author: Filip Tůma
Title: The Replacement Puzzle game
Thesis type: bachelor thesis
Department: Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense: 2019
Study field: Computer Science, full-time form
Supervisor: doc. RNDr. Miroslav Kolařík, Ph.D.
Page count: 31
Supplements: 1 CD
Thesis language: Czech

Anotace

V bakalářské práci byla vytvořena hra Replacement puzzle formou okenní aplikace. Hra dává uživateli možnost vygenerovat novou hru podle počtu tahů, pravidel a znaků a dále uložit a načíst rozehranou hru.

Synopsis

In the bachelor thesis was created game Replacement puzzle in the form of window application. The game gives the user the ability to generate a new game based on the number of moves, rules and characters, and then save and load the game in progress.

Klíčová slova: pravidla, odvozování, C sharp, WPF, SQL

Keywords: replacement puzzle, C sharp, WPF, SQL

Rád bych tímto poděkoval vedoucímu své bakalářské práce doc. RNDr. Miroslavovi Kolaříkovi, Ph.D., za jeho vedení při tvorbě této práce.

Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracoval/a samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.

datum odevzdání práce

podpis autora

Obsah

1	Úvod	7
1.1	Představení hry	7
2	Nástroje použité při implementaci	8
2.1	Jazyk C#	8
2.2	WPF aplikace	8
2.3	SQL	8
3	Programátorská dokumentace	9
3.1	Základní třídy	9
3.1.1	Třída Rule	9
3.1.2	Třída Game	10
3.1.3	Třída User	12
3.2	Design a funkčnost oken	13
3.2.1	StartWindow	13
3.2.2	SignupWindow	14
3.2.3	LoginWindow	15
3.2.4	GameWindow	15
3.2.5	UserGameWindow	19
3.2.6	AccountWindow	20
3.2.7	RulesWindow	20
3.2.8	DialogWindow	20
4	Uživatelská příručka	22
4.1	Spouštění	22
4.2	Herní okno	22
4.2.1	Popis okna	22
4.2.2	Klávesové zkratky	24
4.2.3	Postup ve hře na příkladu	24
4.3	Okno pro tvorbu vlastních her	26
4.3.1	Popis okna	26
4.3.2	Klávesové zkratky	27
	Závěr	28
	Conclusions	29
	A Obsah přiloženého CD	30
	Literatura	31

Seznam obrázků

1	Ukázka hry [1]	7
2	SignupWindow	15
3	GameWindow	16
4	UserGameWindow	19
5	DialogWindow pro zadávání tahů	20
6	DialogWindow pro změnu hesla	21
7	Posuvný panel	22
8	Herní okno	22
9	PopUp box	23
10	Okno s informacemi o uživateli	23
11	Startovní řetězec	24
12	První tah	24
13	Druhý tah	24
14	Třetí tah	25
15	Čtvrtý tah	25
16	Pátý tah	25
17	Okno pro tvorbu vlastních her	26
18	Chybové hlášení	26
19	Možnost hrát hru	27

Seznam tabulek

1	Tabulka oken	14
2	Tabulka User	17
3	Tabulka Move	18

Seznam vět

Seznam zdrojových kódů

1	Metoda generateRule třídy Rule	10
2	Funkce numberOfOccurrences	11
3	Kód hlavního panelu v XAML	13
4	Hashování hesla	14

1 Úvod

1.1 Představení hry

Replacement Puzzle je logická hra, kterou jsem převzal ze stránek prof. Ericha Friedmana¹. Hra obsahuje tři přepisovací pravidla, počáteční a koncovou posloupnost znaků a počet kroků, ve kterých je možné hru dokončit. Pravidla se skládají z levé a pravé strany, kde každá strana je složena z jednoho až tří znaků. Pravidlo lze použít v případě, že posloupnost znaků tvořící jeho levou stranu je obsažena v aktuálním řetězci znaků.

Rule #1: ♠ → ♥♠

Rule #2: ♥♥♥ → ♠♠

Rule #3: ♠♥ → ♠

Goal: ♥♠♠ → ♠♥♥♠♠♠ in 5 moves

♥♠♠ → ♥♥♠♠ → ♥♥♥♠♠ → ♠♠♠♠ → ♠♥♠♠♠ → ♠♥♥♠♠♠
(Rule #1) (Rule #1) (Rule #2) (Rule #1) (Rule #1)

Obrázek 1: Ukázka hry [1]

Pro účely práce je hra pozměněna. Je možnost generovat hry s třemi až pěti pravidly a dvěma až čtyřmi znaky, kterými jsou červené a oranžové kolečko, zelený čtvereček a modrý trojúhelník.

¹Erich Friedman je profesor matematiky na Stetson University v USA. Vytvořil spoustu logických her, mimo jiné právě Replacement puzzle. [1]

2 Nástroje použité při implementaci

V této části textu se stručně zaměřím na implementační nástroje mojí práce, zejména na C#, dále na WPF aplikace a nakonec SQL.

2.1 Jazyk C#

C# je elegantní a typově bezpečný objektově orientovaný jazyk, který umožňuje vývojářům vytvářet různé bezpečné a robustní aplikace, které běží na rozhraní .NET Framework. C# můžete použít k vytvoření tradičních klientských aplikací Windows, webových služeb XML, distribuovaných komponent, aplikací typu klient-server, databázových aplikací a mnoho dalšího. Visual C# poskytuje pokročilý editor kódu, pohodlné návrháře uživatelského rozhraní, integrovaný debugger a mnoho dalších nástrojů, které usnadňují vývoj aplikací založených na verzi 4.0 jazyka C# a verze 4.0 rozhraní .NET Framework. [2]

2.2 WPF aplikace

WPF (Windows Presentation Foundation) je framework pro komplexní tvorbu bohatých formulářových aplikací, který je součástí .NET frameworku od verze 3.0. Disponuje širokou paletou formulářových prvků a také umožňuje bohaté stylování vzhledu aplikace.

Pro vytvoření „uživatelsky bohatého rozhraní“ (RUI) využívá značkovací jazyk XAML, který umožňuje oddělit funkčnost a vzhled aplikace. Cílem WPF je sjednotit uživatelské rozhraní, 2D a 3D grafiku, vektorovou a rastrovou grafiku, animace a provázat s daty audia a videa. [3, 4]

2.3 SQL

Jedná se o dotazovací jazyk používaný pro přístup k informacím v databázi a jejich modifikaci. Některé běžné SQL příkazy zahrnují „insert“, „update“ a „delete“. Jazyk byl poprvé vytvořen IBM v roce 1975 a byl nazýván SEQUEL pro „Structured English Query Language“.

Dnes je SQL běžně používán pro vývoj a správu webových databází. Ačkoliv SQL je nyní považován za standardní jazyk, stále existuje celá řada jeho variant, jako je mSQL a MySQL. Pomocí skriptovacího jazyka, např. PHP, lze příkazy SQL provádět při načítání webové stránky. To umožňuje vytvářet dynamické webové stránky, které mohou zobrazit různé informace při každém načtení. [5]

3 Programátorská dokumentace

V této části se zaměřím na implementaci hry. Nejprve popíšu hlavní třídy a poté se budu věnovat vzhledu a funkčnosti oken.

3.1 Základní třídy

3.1.1 Třída Rule

Třída Rule představuje pravidla. Každé pravidlo obsahuje levou a pravou stranu skládající se ze sekvence dvou až čtyř znaků. V programovací části jsou tyto sekvence reprezentovány jako datové typy string, kde 'k' znamená červené kolečko, 'c' čtvereček, 't' trojúhelník a 'o' oranžové kolečko.

- Predikáty – Třída implementuje metody, respektive predikáty „equals“ a „notRelevant“, které jsou využity při generování nové hry. Metoda „equals“ bere jako argument další pravidlo a vrací *true* v případě, že jsou pravidla shodná; „notRelevant“ vrací *true* právě když levá strana pravidla je rovna straně pravé.
- Konstruktor – Konstruktor třídy Rule využívá metodu „generateRule“, ta bere tři argumenty typu integer, které udávají délku levé, respektive pravé strany pravidla a počet symbolů v pravidle. (Toto je pouze uživatelsky zadané číslo. I když uživatel zadá, že chce pravidla a hru se čtyřmi znaky, tak pravidlo může obsahovat jeden až čtyři znaky.) Metoda pomocí proměnné typu random vygeneruje nejprve levou a poté pravou stranu pravidla.
- Metoda „drawRule“ – slouží k vykreslení pravidel do plátna v herním okně. Jako argument bere proměnnou typu Canvas, jelikož každé pravidlo má pro lepší funkčnost svoje vlastní plátno. Využívá třídy Ellipse pro vykreslení kolečka, Rectangle pro čtvereček, Polygon pro trojúhelník a Line na šipky v pravidlech. Metoda funguje tak, že pro každý znak v pravidle vykreslí jemu odpovídající tvar. Velikost vykreslených znaků je závislá na reálné velikosti plátna v okně. Při každé změně velikosti okna je tím pádem tato metoda vyvolána pro každé pravidlo znovu a dojde k překreslení.
- Metoda „formatToFile“ – slouží pro převedení pravidla do podoby, která se dá uložit v souboru. Jednoduše spojí levou a pravou stranu pravidla a oddělí je znakem 'x'.

```

1 private void generateRule(int l, int r, int symbols)
2 {
3     do
4     {
5         this.left = "";
6         this.right = "";
7         int i = 0;
8         int rnd;
9         while (i < l)
10        {
11            rnd = random.Next(symbols);
12            if (rnd == 0) this.left += "c";
13            else if (rnd == 1) this.left += "k";
14            else if (rnd == 2) this.left += "t";
15            else this.left += "o";
16            i++;
17        }
18
19        i = 0;
20
21        while (i < r)
22        {
23            rnd = random.Next(symbols);
24            if (rnd == 0) this.right += "c";
25            else if (rnd == 1) this.right += "k";
26            else if (rnd == 2) this.right += "t";
27            else this.right += "o";
28            i++;
29        }
30    } while (notRelevant());
31
32 }

```

Zdrojový kód 1: Metoda generateRule třídy Rule

3.1.2 Třída Game

Třída „Game“ znázorňuje hru. Každá hra se skládá ze tří až pěti pravidel, počáteční a koncové posloupnosti.

- Predikáty – Třída obsahuje, obdobně jako třída Rule, predikát „notRelevant“, který vrací hodnotu *true* v případě, že jsou si některá pravidla rovna. Dále predikát „taskEqualsRule“, který je pravdivý, pokud jedno z pravidel odpovídá zadání.
- Konstruktor – bere čtyři argumenty typu integer. První udává minimální počet tahů potřebných k vyřešení hry, druhý udává maximální počet tahů, třetí kolik symbolů se má ve hře vyskytovat a poslední kolik pravidel má hra obsahovat. Metoda v cyklu vždy vytvoří pravidla a poté generuje startovní

řetězec tak dlouho, dokud není obsažena levá strana alespoň jednoho pravidla. V případě, že tato pravidla a startovní řetězec nesplňují podmínky některého z predikátů, je celý průchod cyklem opakován. Pro takto vytvořenou hru je ovšem ještě potřeba vygenerovat cílový řetězec. To probíhá pomocí metody „generateGame“.

- Metoda „generateGame“ je víceméně tou nejdůležitější z pohledu tvorby nové hry. Vytvoří kolekci dvojic klíč – hodnota, obsahující všechny řetězce, které se dají vygenerovat po daném počtu kroků. Tyto řetězce se ukládají do hodnoty dané dvojice. Jako klíč slouží cesta potřebná k vygenerování daného řetězce, tzn. jaká pravidla a na jakém místě byla použita. Na začátku tato kolekce obsahuje pouze startovní posloupnost. V cyklu se pak postupně pro každý řetězec a pro každé pravidlo spočítá počet výskytů levé strany pravidla v daném řetězci pomocí funkce „numberOfOccurrences“. V případě, že výskyt je pouze jeden, tak se levá strana nahradí za pravou a takto vzniklý řetězec je přidán do kolekce (ale lze ho použít až v dalším kroku). Pokud se levá strana v daném řetězci vyskytuje vícekrát, pak je postupně nahrazen každý výskyt. Celý tento proces se opakuje počet tahů-krát. Poté se z kolekce náhodně vybere jedna dvojice jako cílová.

```
1 //spočítá počet výskytů prvního řetězce v druhém řetězci
2 public int numberOfOccurrences(string s, string t)
3 {
4     int counter = 0;
5     int length = t.Length;
6     for (int i = 0; i < length; i++)
7     {
8         if (t.Substring(i, length - i).IndexOf(s) == 0)
9         {
10            counter++;
11        }
12    }
13    return counter;
14 }
```

Zdrojový kód 2: Funkce numberOfOccurrences

- „formatToFile“ je obdobná metoda jako u třídy „Rule“. Převede do reprezentovatelné formy zadání a všechna pravidla, navíc ukládá počet pravidel pro jednodušší načítání hry ze souboru.
- Metoda „drawGame“ – metoda používaná k vykreslení hry do herního okna. Jako argumenty bere plátna, do kterých postupně vykreslí všechna pravidla. Zadání je vykresleno taktéž jako pravidlo.

3.1.3 Třída User

Třída User reprezentuje přihlášeného uživatele v aplikaci. Obsahuje pouze konstruktor nového uživatele na základě zadaného jména a emailu.

3.2 Design a funkčnost oken

Každé z oken má v horní části panel. Tento panel obsahuje název aplikace, tzn. Replacement puzzle, a dále pak v pravém rohu PopUp box, implementovaný pomocí balíčku Material design. Po kliknutí na PopUp box se ukáže StackPanel s dalšími možnostmi. Například u StartWindow je zde pouze možnost zavřít aplikaci.

```
1     <Grid Height="70" VerticalAlignment="Top" Background="
      {DynamicResource {x:Static SystemColors.
      MenuHighlightBrushKey}}" Margin="0,-5,-0.4,0">
2     <Grid.Effect>
3         <DropShadowEffect BlurRadius="1" ShadowDepth="1" Color="#
          FFACACAC"/>
4     </Grid.Effect>
5     <StackPanel Orientation="Horizontal" HorizontalAlignment="
      Right">
6         <materialDesign:PopupBox PlacementMode="
          BottomAndAlignRightEdges" StaysOpen="False" Margin="10"
          >
7             <StackPanel Width="150">
8                 <Button Content="Close" Click="closeBtn_click"/>
9             </StackPanel>
10        </materialDesign:PopupBox>
11    </StackPanel>
12
13    <TextBlock HorizontalAlignment="Center" VerticalAlignment="
      Center" FontSize="50" Foreground="DarkBlue">Replacement
      puzzle</TextBlock>
14 </Grid>
```

Zdrojový kód 3: Kód hlavního panelu v XAML

3.2.1 StartWindow

Okno StartWindow se zobrazí při spuštění aplikace. Kromě hlavního panelu obsahuje ještě pět tlačítek: pro přihlášení a registraci, pro spuštění herního okna bez nutnosti přihlášení, pro zobrazení nápovědy a pro možnost vytvořit si vlastní hru. Kliknutím na jednotlivá tlačítka se vytvoří instance dalších oken.

Tlačítko	Okno
Hrát bez přihlášení	GameWindow
Přihlásit se	LoginWindow
Registrovat se	SignUpWindow
Vytvořit vlastní hru	UserGameWindow
Pravidla	RulesWindow

Tabulka 1: Tabulka oken

3.2.2 SignupWindow

Okno obsahující formulář pro registraci, který se skládá ze čtyř labelů a textboxů pro jméno, heslo a email. Dále se zde nachází tři tlačítka: *Resetovat* pro vymazání textu ve všech textboxech, *Zpět* pro vrácení se zpět do StartWindow a *Potvrdit*. Po kliknutí na toto tlačítko se nejprve zkontroluje, zda je celý formulář validní, tzn., zda text v textboxu pro email je opravdu email a zda se hesla rovnají, a poté provede samotnou registraci. Pokud je registrace provedena úspěšně, zobrazí se uživateli herní okno (GameWindow). V případě neúspěchu se zobrazí chybové hlášení se zprávou, co bylo špatně.

Při registraci se nejprve vygeneruje sůl² a poté je společně s heslem zahashována pomocí algoritmu PBKDF2³. Do databáze se ukládá heslo a sůl společně ve tvaru *sůl + heslo*.

```

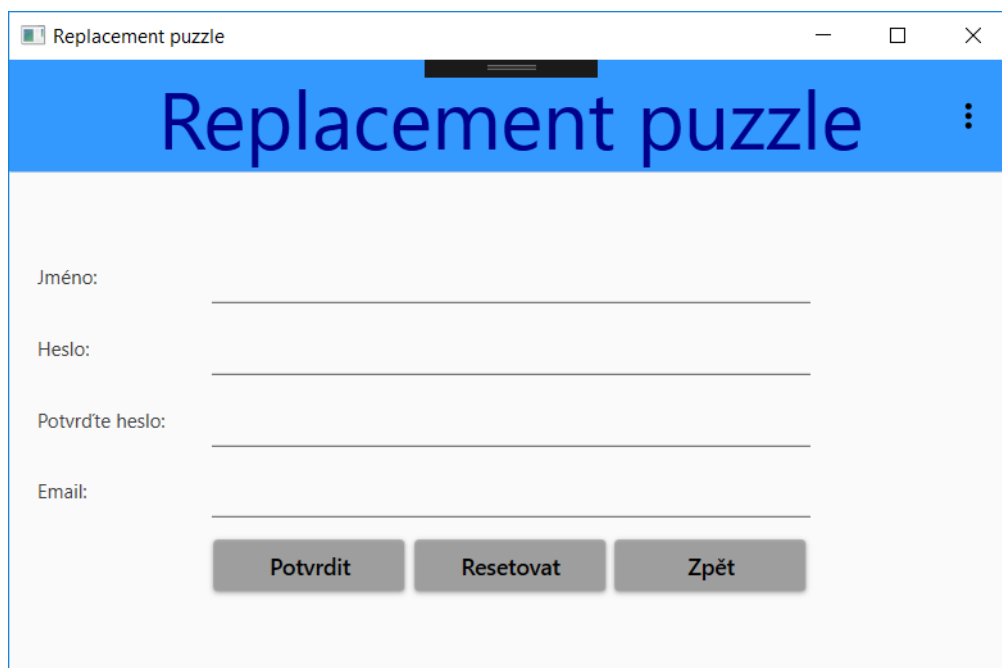
1 //vygeneruje se sůl
2 byte[] salt;
3 new RNGCryptoServiceProvider().GetBytes(salt = new byte[16]);
4
5 //hashování
6 var pbkdf2 = new Rfc2898DeriveBytes(password, salt, 10000);
7 byte[] hash = pbkdf2.GetBytes(20);
8
9 //spojení soli a hashe
10 byte[] hashBytes = new byte[36];
11 Array.Copy(salt, 0, hashBytes, 0, 16);
12 Array.Copy(hash, 0, hashBytes, 16, 20);
13
14 //převedení na string (pro uložení do databáze)
15 string savedPasswordHash = Convert.ToBase64String(hashBytes);

```

Zdrojový kód 4: Hashování hesla

²Sůl je v šifrování řetězec několika znaků, pomocí kterých se rozšíří šifrovaný řetězec.

³PBKDF2 – Password Based Key Derivation Function je kryptografická funkce, která vygeneruje klíč pro zadané heslo a sůl libovolné délky opakovaným voláním pseudonáhodné funkce.



Obrázek 2: SignupWindow

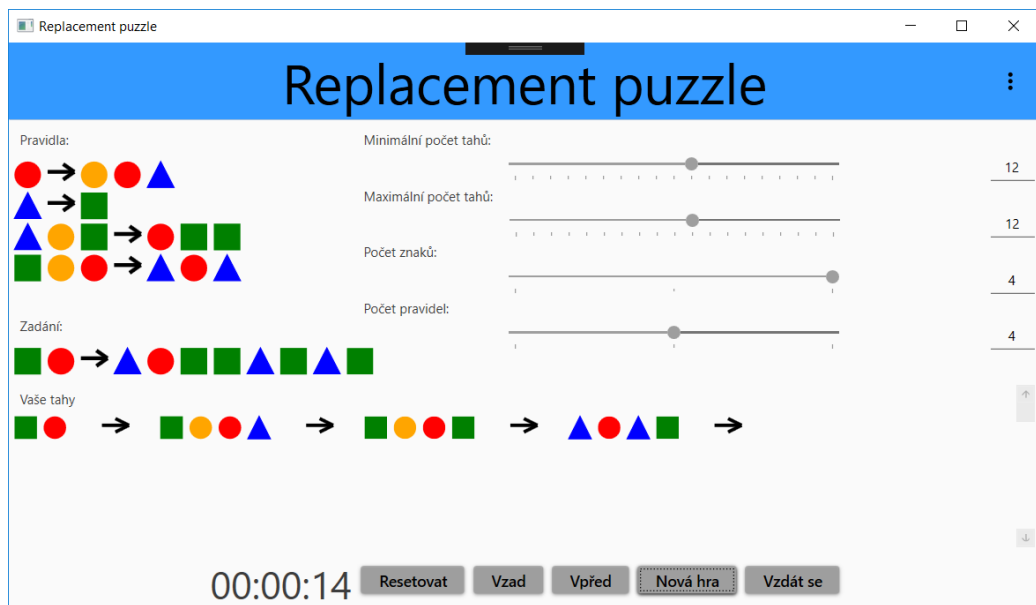
3.2.3 LoginWindow

LoginWindow je ve své podstatě dost podobné SignupWindow. Formulář však obsahuje pouze dva labely a textboxy pro jméno a heslo. Je zde tlačítko *Log in*, které vyvolá metodu pro přihlášení uživatele.

Přihlášení probíhá tím způsobem, že se prohledá databáze, zda se tam nachází uživatel s daným jménem, pokud ano, tak se vezme konkrétní sůl u daného uživatele a pomocí této soli se zahashuje uživatelem napsané heslo. V případě, že se takto získaný hash rovná heslu v databázi, přihlášení proběhne úspěšně a je spuštěno herní okno.

3.2.4 GameWindow

GameWindow je poněkud složitější svým vzhledem. Umožňuje uživateli vygenerovat hru podle zadaného množství tahů, znaků a pravidel. Obsahuje plátina pro vykreslení zadání, pro vykreslení uživatelem provedených tahů a plátina pro každé z pravidel.



Obrázek 3: GameWindow

Základní funkce a události okna:

- Konstruktory okna – jeden konstruktor je bez argumentu a pouze inicializuje všechny komponenty okna a vytvoří dva listy – jeden na provedené tahy a druhý na tahy, ke kterým je možné se dostat pomocí tlačítka „Vpřed“. Druhý konstruktor bere jako jediný argument instanci třídy „Game“. Provede navíc vykreslení dané hry a spuštění časomíry. Druhý konstruktor se používá v případě, že uživatel vytvoří hru v okně „UserGameWindow“ (viz dále) a chce si ji zahrát.
- undoBtn_Click – vymaže poslední provedený tah (pokud byl nějaký tah proveden). Vymazaný tah přidá do listu smazaných tahů.
- redoBtn_Click – opět provede vymazaný tah (pokud byl nějaký tah vymazán). Tento tah je smazán ze seznamu smazaných tahů.
- resetBtn_Click – vymaže všechny dosud provedené tahy bez možnosti návratu pomocí tlačítka „Vpřed“.
- newGameBtn_Click – funkce vezme všechny hodnoty posuvných panelů, pomocí těchto hodnot vytvoří novou hru a zavolá metodu „generateGame“. Po vygenerování hru vykreslí, zpřístupní všechna tlačítka a spustí časomíru.
- Metoda ruleClicked je vyvolána po kliknutí na jednotlivá pravidla nebo při kliknutí na tlačítko vzdát se (viz níže). Jako argumenty bere číslo pravidla a pozici, kde má být pravidlo provedeno. Pozice je v případě kliknutí na pravidlo ignorována (volba pozice je ponechána uživateli). Funkce spočítá

počet výskytů levé strany v aktuálním tahu, pokud je toto číslo rovno nule, pak se ukáže chybové hlášení, které uživateli sdělí, že dané pravidlo nelze použít. Je-li počet výskytů roven jedné, je pravidlo použito, tzn. v aktuálním řetězci je levá strana pravidla nahrazena za pravou a tento řetězec je označen jako nový aktuální řetězec. Je-li počet výskytů větší, zobrazí se „DialogWindow“ s otázkou, na kterém místě má být pravidlo provedeno. Jako odpověď se očekává číslo v rozsahu 1 až délka aktuálního řetězce.

- `giveUpBtn_Click` je vyvolaná po kliknutí na tlačítko *Vzdát se*. Jak již bylo napsáno, při generování hry se všechny možné řetězce ukládají v páru. To je právě kvůli možnosti, že se uživatel vzdá. Tehdy klíč v páru slouží jako jakýsi návod jak se dostat k cílovému řetězci, neboli jaká pravidla v jakém kroku a na jakém indexu použít. První je vždy 0, kterou má jako svůj klíč startovní řetězec, protože k jeho vygenerování není potřeba žádné pravidlo. Další vygenerovaný pár vždy převezme klíč od toho páru, z kterého byl vygenerován a na jeho konec je přidáno číslo pravidla a index místa, kde bylo použito. Tím pádem, když se uživatel vzdá, tak se v cyklu vždy přečtou dva znaky klíče (s tím, že se ignoruje počáteční nula), s nimiž je vyvolána metoda „`ruleClicked`“. Následně je hra vykreslena.
- `checkWin` – kontroluje, zda je poslední provedený tah roven zadanému cílovému řetězci. Pokud ano, vrátí hodnotu `true`.
- `saveToDatabase` – tato funkce je vyvolána v případě, že chce přihlášený uživatel uložit hru. Do databáze jsou ukládány informace o uživateli, informace o rozehrané hře a všechny provedené tahy. Ukládání probíhá ve dvou fázích, nejprve je uložena samotná hra do tabulky *Game* a poté se spustí cyklus přes všechny provedené tahy a tyto tahy se ukládají do tabulky *Move*
 - Tabulka *User* ukládá uživatelské id (autoincrement), jméno, email a jeho heslo v zahashované podobě.

id	name	email	passwordHash
6	Marek	ma.rek@gmail.com	QhQqF6QHsrNIU3I...
7	Filip	filip@tuma.cz	07pS5dAZbxN7ooc...

Tabulka 2: Tabulka *User*

- Tabulka *Game* je rozsáhlejší. Ukládá celkem 13 položek:
 - * `userID` – ID uživatele, který tuto hru uložil
 - * `time` – čas na časomíře v době uložení
 - * `name` – jméno uložené hry
 - * `first` – první pravidlo

- * second – druhé pravidlo
- * third – třetí pravidlo
- * fourth – čtvrté pravidlo
- * fifth – páté pravidlo
- * minMoves – minimální počet tahů potřebných k řešení
- * maxMoves – maximální počet tahů potřebných k řešení
- * start – počáteční řetězec
- * end – koncový řetězec
- * id – id hry

Pravidla jsou uložena ve tvaru *levá strana + znak 'x' + pravá strana*. Čtvrté a páté pravidlo může nabývat hodnoty null, protože hra tato pravidla nemusí obsahovat. Koncový řetězec je pak uložen ve tvaru *klíč + znak 'x' + hodnota*.

– Tabulka Move obsahuje celkem pět položek:

- * nom – integer, udávající v jakém kroku byl tah proveden
- * userID – id uživatele, který si tuto hru, respektive tah uložil
- * move – samotný řetězec reprezentující tah
- * gameName – jméno hry
- * id – id tahu

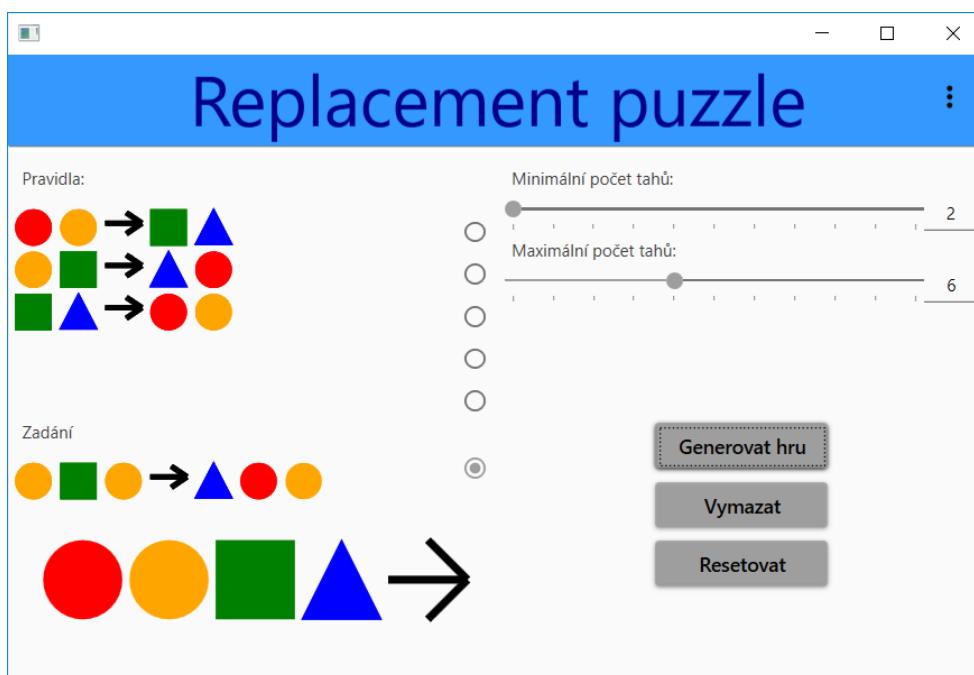
nom	userID	move	gameName	id
1	3	kctt	game02	9
2	3	kcttct	game02	10
3	3	kkttct	game02	11
4	3	kkttkt	game02	12

Tabulka 3: Tabulka Move

- saveToFile – ukládá hry do souboru. Tato metoda se použije v případě, že uživatel není přihlášený. Pro uložení do souboru se používají metody „formatToFile“ třídy *Rule a Game*, které převedou hru do potřebné podoby. Struktura souboru je velice podobná struktuře tabulky v databázi s tím rozdílem, že do souboru je uložen i počet pravidel, pro jednodušší načítání hry.
- loadFromDatabase – slouží k načítání hry z databáze. Ve své podstatě je to opak funkce saveToDatabase.
- loadFromFile – slouží k načítání hry ze souboru.

3.2.5 UserGameWindow

UserGameWindow opět obsahuje plátna na vykreslení pravidel a zadání, ke kterým náleží skupina přepínačů. Dále se v okně nachází plátno, nesoucí symboly, které slouží jako tlačítka.



Obrázek 4: UserGameWindow

- Událost `toolCanvasMouseDown` – je vyvolána po kliknutí na plátno nesoucí symboly v dolní části okna. Nejprve určí symbol, na který bylo kliknuto. To se provede jednak podle třídy, do které symbol náleží (`Ellipse`, `Polygon`, `Rectangle`, `Line`), v případě koleček se dále rozeznává barva. Poté se podle zaškrtnutého přepínače určí pravidlo, do kterého má být symbol vykreslen. V případě, že není zaškrtnutý žádný, uživateli se ukáže chybové hlášení. Jestliže je zaškrtnutý nějaký přepínač, tak se zkontroluje, zda je v jemu odpovídajícím pravidle místo (jelikož levá i pravá strana může obsahovat 1–3 znaky, tak maximální délka je včetně šipky sedm znaků), pokud ano, je v kódu přidán do řetězce reprezentujícího dané pravidlo znak, odpovídající symbolu (písmena *o*, *c*, *k*, *t*) a plátno je překresleno. Symboly tak nejsou přidávány přímo do plátna. To je z důvodu lepší práce s pravidly, kdy se snadněji maže poslední přidávaný symbol.
- `deleteBtn_Click` – vymaže poslední znak ve vybraném pravidle.
- `resetBtn_Click` – vymaže všechny uživatelem zadané symboly.

- generateBtn_Click – vytvoří novou hru s pravidly zadanými uživatelem a pomocí metody „generateGame“ vygeneruje seznam obsahující všechny řetězce, které se dají dosáhnout v uživatelem zadaném rozmezí tahů. V případě, že v tomto seznamu je obsažen i zadaný cílový řetězec, je vrácena hodnota určující, v kolikátém tahu je hra splnitelná a uživateli je to oznámeno pomocí DialogWindow, které navíc nabídne možnost si takhle vytvořenou hru zahrát.
- isRelevant – kontroluje, zda uživatelem zadaná pravidla a zadání jsou korektní. Tzn. jestli poslední zadaný symbol v kterémkoliv pravidle nebo zadání není šipka, případně jestli je v pravidlech šipka vůbec obsažena. Dále také kontroluje, zda jsou zadána alespoň tři pravidla.

3.2.6 AccountWindow

Okno, které ukazuje informace o přihlášeném uživateli (jméno a email). Zároveň mu dává možnost tyto informace a heslo změnit pomocí událostí:

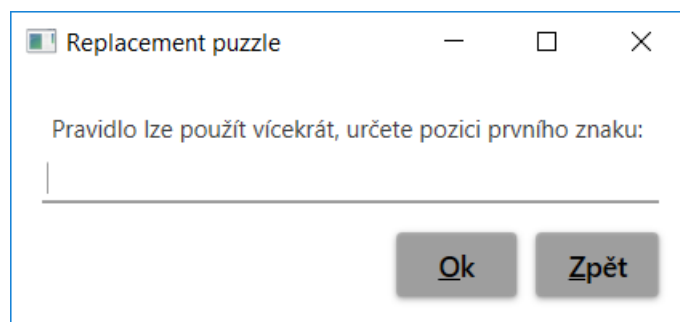
- changePasswordBtn_Click
- changeEmailBtn_Click
- changeNameBtn_Click

Všechny tři pracují s databází. Změna jména a emailu je provedena přímo, u změny hesla se nejprve zkontroluje, zda zadané staré heslo odpovídá heslu v databázi.

3.2.7 RulesWindow

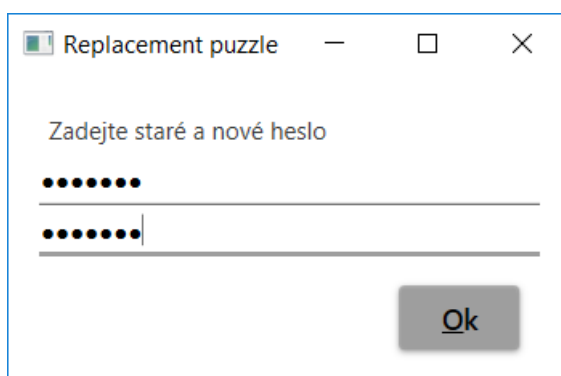
Okno obsahující pravidla hry a základní ovládání.

3.2.8 DialogWindow



Obrázek 5: DialogWindow pro zadávání tahů

Toto okno obsahuje *label* pro možnost zadání otázky, či požadavku pro uživatele, dále jeden *textbox* a dva *passwordboxy* pro zadávání hesla. Jejich viditelnost se nastavuje dle potřeby práce s dialogovým oknem. Okno v podobě na obrázku 5 se používá v případě, že pravidlo lze použít na více místech v řetězci. Okno na obrázku 6 se používá při změně hesla.



Obrázek 6: DialogWindow pro změnu hesla

4 Uživatelská příručka

4.1 Spouštění

Aplikaci lze spustit přímo z příloženého CD jako soubor .exe v adresáři bin/.

4.2 Herní okno

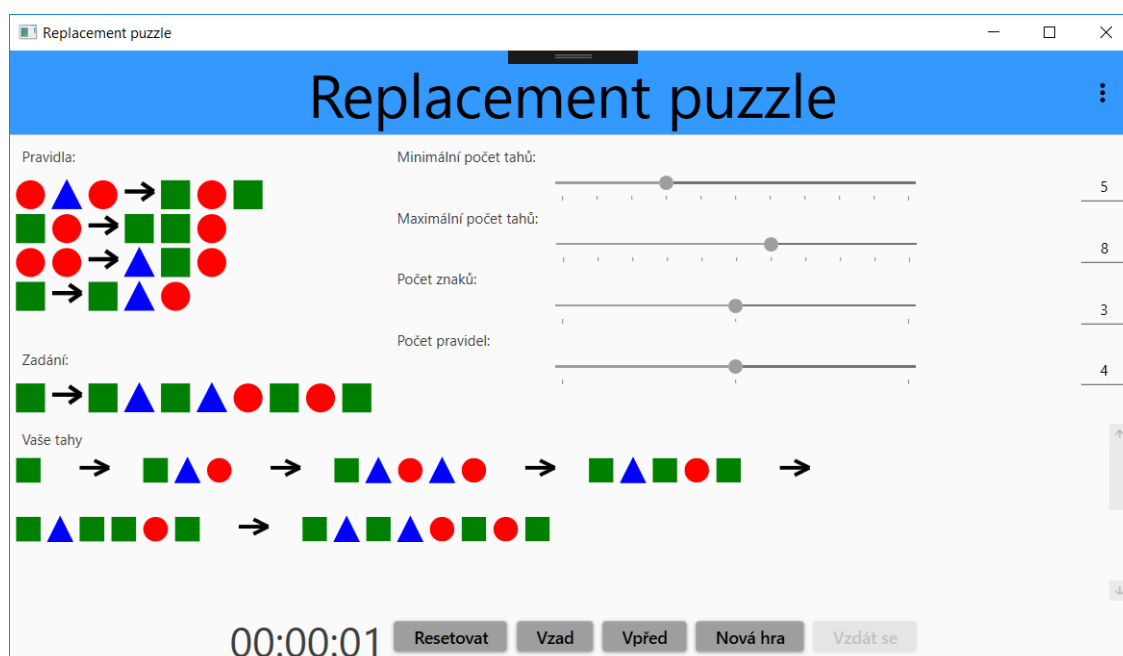
4.2.1 Popis okna

Okno obsahuje posuvné panely pro nastavení minimálního a maximálního počtu tahů, počtu znaků a pravidel. Hodnota na každém z panelů jde zadat i pomocí čísla nacházejícího se napravo od panelu.



Obrázek 7: Posuvný panel

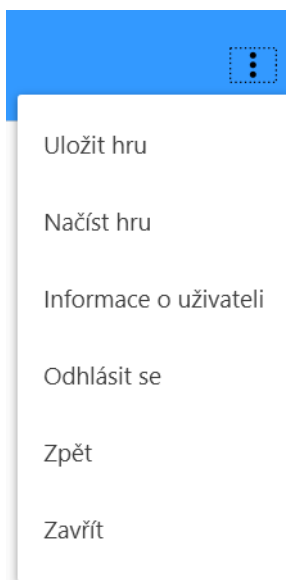
Po zadání hodnot na všech panelech se kliknutím na tlačítko „Nová hra“ vygeneruje hra a vykreslí se pravidla a zadání. Tato pravidla se nachází po levé straně okna. Pravidla je možné použít kliknutím myši na konkrétní pravidlo, nebo pomocí klávesových zkratk (viz dále).



Obrázek 8: Herní okno

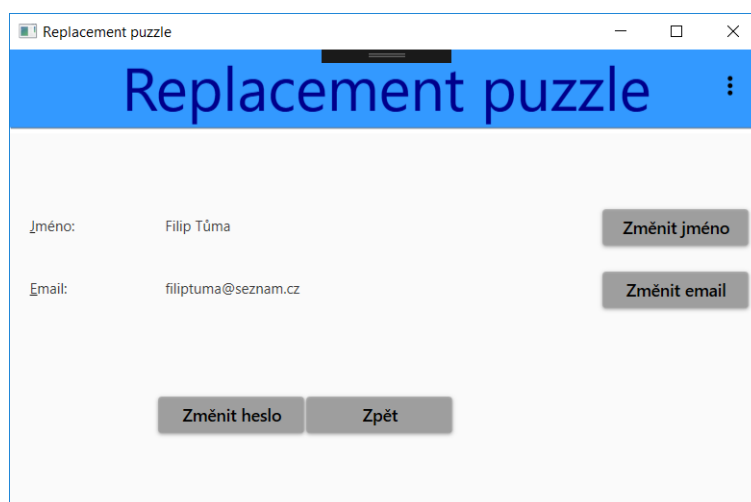
Všechny vaše tahy se vykreslují do plátna v dolní části okna. Pokud je tahů hodně a nevezou se do plátna, je umožněno plátnem pohybovat nahoru a dolů pomocí posuvníku na pravé straně plátna.

V pravé části hlavního panelu se nachází „PopUp box“, po kliknutí na něj se ukáže nabídka s možnostmi.



Obrázek 9: PopUp box

Možnost *Odhlásit se* a *Informace o uživateli* se zobrazí jen přihlášeným uživatelům. Tlačítko *Informace o uživateli* otevře okno, ve kterém se zobrazí jméno a email uživatele a umožní tyto informace změnit.



Obrázek 10: Okno s informacemi o uživateli

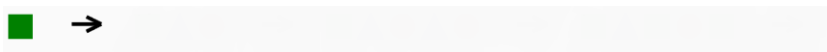
4.2.2 Klávesové zkratky

Pro ovládání herního okna se dají použít následující zkratky:

- Ctrl + S – uložení hry
- Ctrl + O – načtení uložené hry
- Šipka doleva – návrat o tah zpět (tlačítko „Vzad“)
- Šipka doprava – pro opětovné provedení zrušeného tahu (tlačítko „Vpřed“)
- Delete – smazání všech provedených tahů (tlačítko „Resetovat“)
- N – vygenerování nové hry (tlačítko „Nová hra“)
- SPACE – uživatel se vzdává (tlačítko „Vzdát se“)
- Ctrl + 1-5 – použití jednotlivých pravidel (dle čísla)

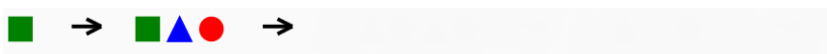
4.2.3 Postup ve hře na příkladu

Pro pravidla a zadání hry na obrázku 8 by řešení mohlo vypadat následovně:



Obrázek 11: Startovní řetězec

Na začátku máme pouze počáteční řetězec. Zde nemáme na výběr a musíme použít čtvrté pravidlo.



Obrázek 12: První tah

Pro druhý tah rovněž nemáme možnost výběru a musíme opět použít čtvrté pravidlo.



Obrázek 13: Druhý tah

Pro třetí tah máme dvě možnosti, použít opět čtvrté pravidlo, nebo použít pravidlo první. My se rozhodneme pro první pravidlo.



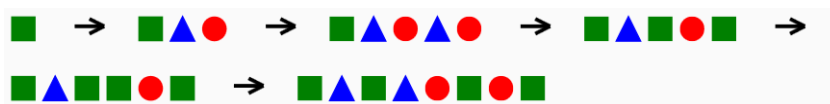
Obrázek 14: Třetí tah

Pro čtvrtý tah máme možností ještě více. Čtvrté pravidlo lze použít třikrát a navíc lze použít druhé pravidlo. V tomto případě použijeme pravidlo druhé.



Obrázek 15: Čtvrtý tah

Pro pátý tah lze opět čtvrté pravidlo použít několikrát, my zvolíme druhý čtvereček, tedy třetí symbol v pravidle.



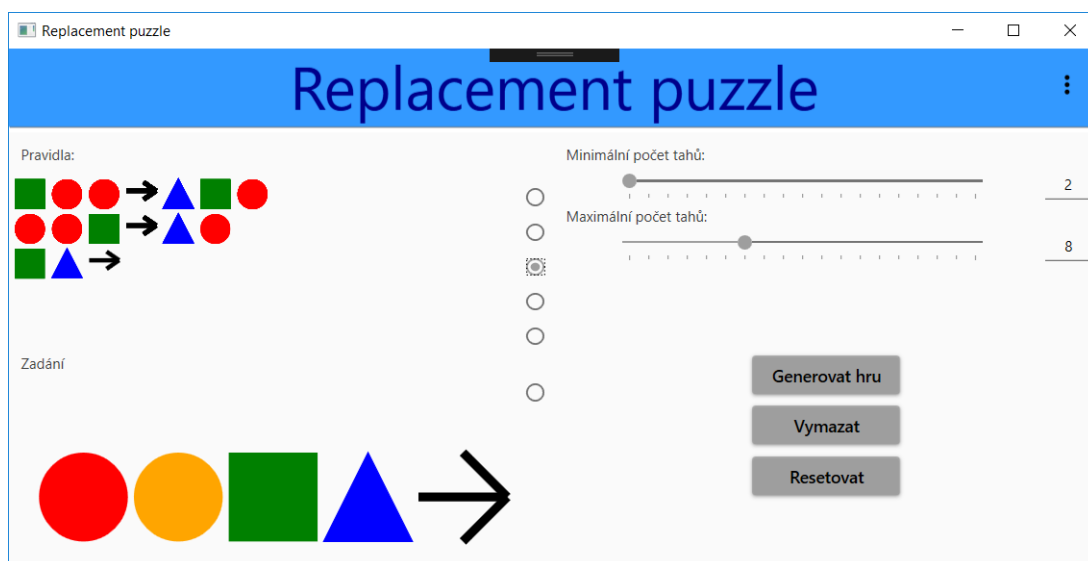
Obrázek 16: Pátý tah

Po provedení pátého tahu jsme úspěšně splnili zadání. Tento příklad byl pouze demonstrativní. V každém kroku jsme proto provedli ten správný tah, vedoucí k řešení hry. Pro případ provedení tahu, který vedl do slepé uličky, okno obsahuje tlačítka „Vzad“ a „Resetovat“, jejichž funkčnost byla zmíněna v části *Klávesové zkratky*.

4.3 Okno pro tvorbu vlastních her

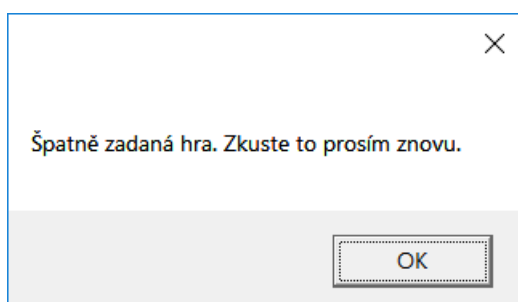
4.3.1 Popis okna

Okno opět obsahuje posuvné panely. Tentokrát pouze dva – na minimální a maximální počet tahů. Počet znaků a pravidel určuje uživatel sám. V levé spodní části okna se nacházejí kolečka, čtverec, trojúhelník a šipka, které slouží jako tlačítka. Pokud je označen některý přepínač, pak kliknutím na jeden ze zmíněných objektů je tento objekt přidán do pravidla označeného konkrétním přepínačem.



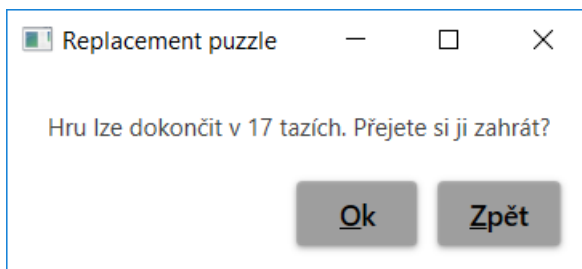
Obrázek 17: Okno pro tvorbu vlastních her

Při zadávání pravidel je potřeba dbát na to, aby posledním symbolem nebyla šipka, v takovém případě je při generování hry vyvoláno chybové hlášení. (Aby byla šipka první není vůbec umožněno.) Zároveň musí být zadaná minimálně tři pravidla.



Obrázek 18: Chybové hlášení

Po kliknutí na „Generovat hru“ jsou dvě možnosti: zadaná hra je nebo není řešitelná v daném rozmezí tahů. V obou případech je to oznámeno uživateli. Pokud hra řešitelná je, pak je uživateli oznámeno v kolika krocích lze hra vyřešit a je mu dána možnost si hru zahrát.



Obrázek 19: Možnost hrát hru

4.3.2 Klávesové zkratky

Obdobně jako v herním okně, i v tomto okně lze využít klávesových zkratk:

- G – vygenerování nové hry
- Backspace – smazání posledního znaku v označeném pravidle (tlačítko „Vymazat“)
- Delete – smazání všech znaků ze všech pravidel (tlačítko „Resetovat“)
- A – přidání šipky do označeného pravidla
- C – přidání čtverečku do označeného pravidla
- K – přidání červeného kolečka do označeného pravidla
- T – přidání trojúhelníku do označeného pravidla
- O – přidání oranžového kolečka do označeného pravidla

Závěr

V práci byla vytvořena hra Replacement puzzle formou okenní aplikace. Hra Vám umožňuje dle zadání generovat nové hry, ukládat rozehrané hry a tyto hry opět spouštět. Oproti verzi ze stránek prof. Friedmana byla hra rozšířena o možnost volby počtu pravidel, znaků a tahů při generování nových her. Snažil jsem se aplikaci psát dostatečně obecně tak, aby šlo případně přidat další tahy a znaky, např. hvězdu.

Aplikace by mohla sloužit jako pomůcka pro pochopení fungování substituce či používání odvozovacích pravidel. Dalším možným využitím je procvičování logického myšlení.

Conclusions

Replacement puzzle game was created in the form of window application. The game allows you to generate new games, save games in progress, and run these games again. Compared to prof. Friedman's game was expanded to include a choice of rules, characters and moves to generate new games. I tried to write the application in general enough to add additional moves and characters, such as a star.

The application could serve as an aid to understanding the functioning of substitution or the use of inference rules. Another possible use is to practice logical thinking.

A Obsah příloženého CD

V této sekci je uveden stručný popis obsahu příloženého CD, tj. jeho závazné adresářové struktury, důležitých souborů apod.

bin/

Program Replacement puzzle spustitelný přímo z CD.

doc/

Text práce ve formátu PDF, vytvořený s použitím závazného stylu KI PřF UP v Olomouci pro závěrečné práce, včetně všech příloh, a všechny soubory potřebné pro bezproblémové vygenerování PDF dokumentu textu (v ZIP archivu), tj. zdrojový text textu, vložené obrázky, apod.

src/

Kompletní zdrojové kódy a další soubory potřebné pro bezproblémové spuštění programu.

readme.txt

Instrukce pro instalaci a spuštění programu, včetně všech požadavků pro jeho bezproblémový provoz.

Literatura

- [1] Erich Friedman - logické hry. Dostupné z:
<https://www2.stetson.edu/~efriedma/puzzle/index.html>
- [2] C Sharp. Dostupné z:
[https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2012/z1zx9t92\(v=vs.110\)](https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2012/z1zx9t92(v=vs.110))
- [3] Windows Presentation Form - Wikipedie. Dostupné z:
https://cs.wikipedia.org/wiki/Windows_Presentation_Foundation
- [4] Windows Presentation Form. Dostupné z:
<https://www.itnetwork.cz/csharp/formulare/wpf/>
- [5] SQL - Structured Query Language. Dostupné z:
<https://techterms.com/definition/sql>