

Tahová strategická hra v internetovém prohlížeči

Turn based strategic game in web browser

Bc. Libor Tichý

Diplomová práce
2011



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Libor TICHÝ**
Osobní číslo: **A09430**
Studijní program: **N 3902 Inženýrská informatika**
Studijní obor: **Počítačové a komunikační systémy**

Téma práce: **Tahová strategická hra v internetovém prohlížeči**

Zásady pro vypracování:

1. Navrhněte ve spolupráci s vedoucím práce herní mechanismy.
2. Vypracujte herní engine v programovacím jazyce PHP.
3. Umožněte hru proti počítači.
4. Využijte objektové programování (PHP framework - Nette) a javascript framework jQuery.
5. Součástí práce není grafické zpracování bitmapových podkladů.

Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. Gamedesign.cz [online]. Vývoj počítačových her obecně. Dostupné z WWW: [http://gamedesign.cz].
2. Rutledge, M. PHP game programming. Thoson course technology. ISBN: 1-59200-153-X.
3. Morrison, M. Naučte se programovat počítačové hry za 24 hodin. ComputerPress, 2004, ISBN 80-251-0371-4.
4. Top 1 to 50 Games and Entertainment - PHP scripts at Programsdb.com [online]. Game scripts. Dostupné z WWW: [http://www.programsdb.com/scriptscategory/10/44/1/0.html].
5. Straffin, P.D.: Game Theory and Strategy, The Mathematical Association of America, 2003.
6. Dresher, M.: The Mathematics of Games of Strategy, Theory and Applications, Dover Publications, 1981.
7. Hýkšová, M. Teorie her a optimální rozhodování, výukové materiály [online], Dostupné z WWW:[http://euler.fd.cvut.cz/predmety/teorie_her].

Vedoucí diplomové práce:

doc. Ing. Martin Sysel, Ph.D.

Ústav počítačových a komunikačních systémů

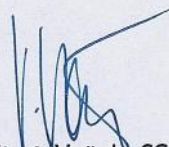
Datum zadání diplomové práce:

25. února 2011

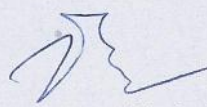
Termín odevzdání diplomové práce:

13. června 2011

Ve Zlíně dne 25. února 2011


prof. Ing. Vladimír Vašek, CSc.
děkan




prof. Ing. Karel Vlček, CSc.
ředitel ústavu

ABSTRAKT

Cílem této diplomové práce je navrhnout a vytvořit online tahovou strategickou hru, kterou lze hrát ve webovém prohlížeči. Práce zahrnuje úvod do teorie her, kde jsou představeny základní pojmy z této oblasti, rozebrány klasifikace her a modely jejich zápisu. V textu jsou dále diskutovány technologie, které byly použity pro tvorbu aplikace na straně serveru a popsána architektura softwaru Model-View-Controller. Část práce je věnována popisu herních mechanismů a pravidlům hry. Objektově orientovaný model aplikace je vyjádřen pomocí grafického jazyka UML. Závěrem jsou rozebírána některá programová řešení.

Klíčová slova: tahová strategická hra, teorie her, Nette framework, architektura MVC

ABSTRACT

The aim of this thesis is to design and develop an online turn based strategic game in web browser. The text includes an introduction to game theory, which introduces basic terms of this field, discuss the classification of games and models for their notation. Text is also focused for technologies that were used to build the application on the server and describes the software architecture Model-View-Controller. Part of the work is devoted to describing the mechanism and rules of the game play. Object-oriented application design is written using a graphical language UML. Finally are discussed some of the solutions that were used in program.

Keywords: turn based strategic game, game theory, Nette framework, MVC architecture

Rád bych na tomto místě poděkoval doc. Ing. Martinu Syslovi Ph.D. za trpělivost a čas, který mi věnoval, za cenné věcné rady a připomínky. Také děkuji za podporu své rodině, zvláště mému bratru Antonínovi. Děkuji také Zbyňku Máčalovi, který pro tuto hru vytvořil a poskytl modely zbraní.

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....
podpis diplomanta

OBSAH

ÚVOD.....	10
I TEORETICKÁ ČÁST	11
1 TEORIE HER.....	12
1.1 POJMY	12
1.2 ZÁKLADNÍ MODELY PRO VYJÁDŘENÍ HRY	13
1.2.1 Hra v explicitním tvaru	13
1.2.2 Hra v normálním tvaru	14
1.3 KLASIFIKACE HER.....	15
1.3.1 Podle počtu účastníků (hráčů).....	15
1.3.2 Podle velikosti prostoru strategií.....	15
1.3.3 Podle zájmu hráčů	15
1.3.4 Podle typu výhry.....	15
1.3.5 Podle informací o důsledcích volby	15
1.4 MATICOVÉ HRY	16
1.5 ANTAGONISTICKÉ HRY	16
2 POUŽITÉ TECHNOLOGIE.....	18
2.1 PROGRAMOVACÍ JAZYK	18
2.2 NETTE FRAMEWORK.....	18
2.3 MVC.....	19
2.4 DATABÁZOVÝ SYSTÉM	20
2.4.1 MyISAM	21
2.4.2 InnoDB	21
2.5 NOTORM.....	22
2.6 JQUERY	23
2.6.1 jQuery PLUGINS	23
2.6.2 jQuery UI.....	23
3 HRY V INTERNETOVÉM PROHLÍŽEČI	25
3.1 HRY NA STRANĚ SERVERU	25
3.2 HRY NA STRANĚ KLIENTA	25
II PRAKTICKÁ ČÁST	27
4 HERNÍ MECHANISMY HRY MĚSTO ZLODĚJŮ	28

4.1	POPIS HRY	28
4.2	MAPA MĚSTA.....	28
4.3	ZAČÁTEK HRY	30
4.4	BANKA	30
4.5	CASINO.....	30
4.6	OBCHOD SE ZBRANĚMI	30
4.6.1	Přehled zbraní.....	31
4.7	POSTAVA.....	35
4.7.1	Útočné vlastnosti	35
4.7.2	Obranné vlastnosti.....	35
4.8	ULICE	36
4.8.1	Přepadení.....	36
5	HRA GOLDTIP.....	37
5.1.1	Partie hry v normálním tvaru.....	37
5.1.2	Příklad hry	38
5.1.3	Vytvoření hry.....	39
5.1.4	Připojení ke hře	39
5.1.5	Hra proti počítači.....	39
6	APLIKACE.....	41
6.1	PŘÍPADY UŽITÍ.....	41
6.1.1	Registrace účtu	41
6.2	E-R MODEL DATABÁZE.....	42
6.3	ROZDĚLENÍ NA PRESENTERY.....	43
6.3.1	BasePresenter	44
6.3.2	MainpagePresenter	44
6.3.3	RegistracePresenter	45
6.3.4	AdminPresenter.....	45
6.3.5	MovesPresenter	45
6.3.6	BankPresenter.....	45
6.3.7	CasinoPresenter.....	45
6.3.8	GoldtipPresenter.....	46
6.3.9	ShopPresenter.....	46
6.3.10	StreetPresenter.....	46
6.3.11	CrimesPresenter	46
6.4	MODEL.....	47
6.4.1	Třída Acl	47
6.4.2	Třída Logs	47
6.4.3	Třída Account.....	48
6.4.4	Třída Character.....	48
6.4.5	Třída Weapon.....	49
6.4.6	Třída Crime	50
6.4.7	Třída Goldtip.....	51
6.4.8	Třída Goldtips	53

6.4.9	Třída Place.....	53
6.4.10	Třída Places.....	54
6.5	TÝDENNÍ PLAT.....	55
6.6	POZICE POSTAVY.....	55
6.7	POHYB POSTAVY.....	56
1.	Cron.....	57
2.	MySQL Events.....	58
ZÁVĚR.....		59
ZÁVĚR V ANGLIČTINĚ.....		60
SEZNAM POUŽITÉ LITERATURY.....		61
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....		63
SEZNAM OBRÁZKŮ.....		64
SEZNAM TABULEK.....		65
SEZNAM PŘÍLOH.....		66

ÚVOD

S rozvojem internetu a technologií pro tvorbu internetových aplikací se rozvíjí také nemalé softwarové odvětví – internetové hry. Do popředí se dostávají hry, které umožňují hru dvou a více hráčů, kde je možné vytvářet koalice a spolupracovat. Hráči, které jsou vytvářeny pro webové prohlížeče, se dostalo velké popularity, neboť u nich není třeba instalovat žádný dodatečný software. Ke zprovoznění stačí webový prohlížeč, většinou s podporou JavaScriptu nebo technologie Flash.

K nejrozšířenějším jazykům pro tvorbu webových aplikací patří jazyk PHP, který běží na straně serveru a obsahuje mnoho základních funkcí pro práci s databázemi, obrázky, textem atd. Různé aplikace mají určité části a funkce stejné, nevyplatí se je programovat pokaždé znovu, proto vznikly frameworky jako nástavba PHP (Nette, Zend, Ruby On Rails a další). Tvorba aplikace je tak mnohonásobně rychlejší.

Samotný návrh hry a herních mechanismů je klíčový pro to, aby hra byla úspěšná a hratelná. Základem je mít dobrý nápad, což vyžaduje kreativitu tvůrců a také určité zkušenosti. Analýzou her, strategiemi a jejich popisem se zabývá široký obor teorie her a své uplatnění získal ve vojenství a ekonomii.

Záměrem mé práce bylo navrhnout a vytvořit strategickou tahovou hru ve webovém prohlížeči. Vytvořená hotová aplikace je přiložena na kompaktním disku.

I. TEORETICKÁ ČÁST

1 TEORIE HER

Pomocí základních pojmů (Kapitola 1.1) můžeme definovat předmět teorie her následovně: Teorie her zkoumá způsoby určení strategie (optimální strategie), se kterou dosáhne účastník rozhodovací situace v partii co nejvyšší výhry (efektu). [13]

Základním pojmem, který zkoumá teorie her, je tedy rozhodovací situace. V této situaci se může vyskytnout jeden nebo i více rozhodovatelů, rozhodovatele budeme dále nazývat hráči. Pokud se hráči svými rozhodnutími navzájem neovlivňují, neboli nedochází ke konfliktu zájmů, lze celou situaci řešit pomocí lineárního programování. Pokud se však vyskytnou v rozhodovací situaci dva nebo i více hráčů, může nastat konflikt zájmů velmi lehce, a to tím, že rozhodnutí jednoho hráče může být pro ostatní nevýhodné, dokonce jim může uškodit. Zde už tedy nestačí zjistit správná řešení pouze pro jednoho hráče, ale je třeba hledat řešení tak, aby byl výsledek výhodný pro všechny účastníky.

1.1 Pojmy

Hra je soubor všech pravidel a podmínek, kterými je určen prostor strategií. Hrou také rozumíme matematický model rozhodovací situace, nazývaný též jako hra v normálním tvaru (Kapitola 1.2.2). Grafický model se nazývá hra v explicitním tvaru (Kapitola 1.2.1).

Hráč je účastník hry, který svým chováním může ovlivnit její výsledek. Hráč může být buď inteligentní, který logicky analyzuje konfliktní situaci a volí strategie tak, aby maximalizoval svou výhru, nebo neinteligentní, který volí svá rozhodnutí podle nějakého rozložení pravděpodobnosti a nezáleží mu při tom na hodnotě vlastní výhry.

Partií hry rozumíme realizaci hry.

Tahem nazveme moment hry, ve kterém hráči uskuteční volbu jedné strategie z prostoru strategií. Partie je složena buď z jednoho tahu každého hráče, nebo z posloupnosti více tahů.

Výplatní funkce je předpis (funkce), který vyhodnotí důsledek volby strategie.

Výhrou označíme hodnotu výplatní funkce, je to také míra užitku, kterou hráči přinese jeho rozhodnutí. Míra užitku může být vyjádřena penězi, body nebo pocitem uspokojení a může být kladná i záporná.

Prostor strategií je množina všech přípustných řešení každého hráče. Má-li tedy hráč volit tu nejvhodnější strategii, musí být schopen určit, jakou výhru mu každá strategie přinese.

Optimální strategie je taková, která zaručuje hráči tu nejvyšší výhru nezávisle na tom, jakou strategii zvolí protihráč.

1.2 Základní modely pro vyjádření hry

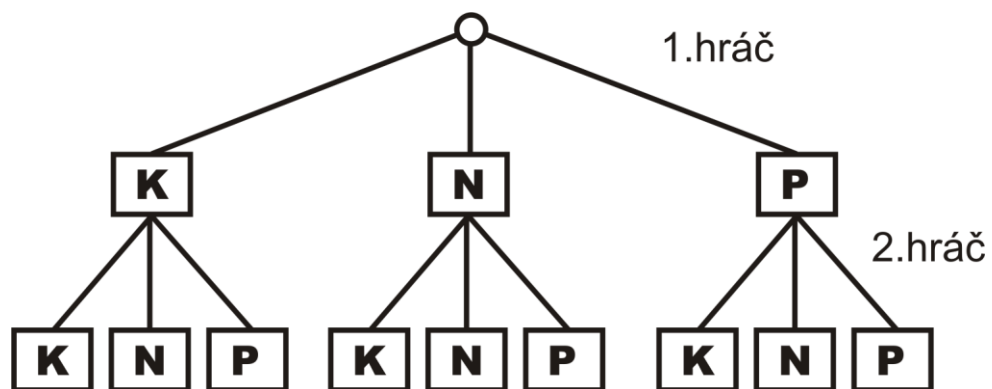
Model hry nebo konfliktní situace musí být pečlivě sestaven tak, aby co nejlépe popisoval daný problém. Příliš jednoduchý model může vést k zanedbání některých důležitých aspektů reálné hry. Naproti tomu příliš složitý model může díky nepřehlednosti znesnadnit nalezení stěžejních bodů hry. Aby se tedy vyhnulo těmto dvěma extrémům, existuje několik forem, jak situaci popsat. K těm nejdůležitějším patří vyjádření hry v *explicitním* nebo *normálním* tvaru. [10, 7]

1.2.1 Hra v explicitním tvaru

Matematický model rozhodovací situace, v níž rozhodnutí jednotlivých hráčů probíhá ve formě postupně prováděných tahů (etap). [7]

Model lze znázornit tzv. stromem hry. Strom hry zachycuje všechny situace, které ve hře mohou nastat. Každé situaci odpovídá jeden uzel, z něhož vychází určitý počet hran odpovídajících možným rozhodnutím (tahům).

Příklad hry Kámen-nůžky-papír v explicitním tvaru je na Obrázku 1, kámen je označen písmenem K, nůžky písmenem N a papír P.



Obrázek 1: Příklad hry Kámen-nůžky-papír v explicitním tvaru

1.2.2 Hra v normálním tvaru

Definice 1 dle [7]

Nechť je dána konečná neprázdná n -prvková množina $Q = \{1, 2, \dots, n\}$, n množin S_1, S_2, \dots, S_n a n reálných funkcí u_1, u_2, \dots, u_n definovaných na kartézském součinu $S_1 \times S_2 \times \dots \times S_n$.

Hrou n hráčů v normálním tvaru se rozumí uspořádanou $(2n + 1)$ -tici.

$$\{Q, S_1, \dots, S_n; u_1(s_1, \dots, s_n), \dots, u_n(s_1, \dots, s_n)\} \quad (1)$$

Přičemž Q je množina hráčů, S_i množina strategií hráče i , prvek $s_i \in S_i$ je strategie hráče i a funkce $u_i(s_1, \dots, s_n)$ výplatní funkce hráče i . Hra v normálním tvaru umožňuje zkoumat možné strategie a výhry jednotlivých hráčů.

Příklad modelu hry Kámen-nůžky-papír je uveden na Obrázku 2, označení jsou stejná jako u příkladu s explicitním tvarem (Obrázek 1). Vítězství je označeno hodnotou 1, prohra -1 a remíza hodnotou 0.

		<i>hráč 2</i>		
		K	N	P
<i>hráč 1</i>	K	0	1	-1
	N	-1	0	1
	P	1	-1	0

Obrázek 2: Příklad hry Kámen-nůžky-papír v normálním tvaru

1.3 Klasifikace her

Hry jsou klasifikovány z různých hledisek.

1.3.1 Podle počtu účastníků (hráčů)

- Hry *dvou hráčů*
- Hry *N hráčů*, kde $N > 2$

U her s více než dvěma hráči může docházet ke vzniku tzv. koalic.

1.3.2 Podle velikosti prostoru strategií

- Hry s konečným počtem strategií
- Hry s nekonečným počtem strategií

1.3.3 Podle zájmu hráčů

- Antagonistické hry
- Neantagonistické hry

1.3.4 Podle typu výhry

- Hry s konstantním součtem
- Hry s nekonstantním součtem

Pro hry s konstantním součtem platí, že součet výher všech hráčů je konstantní pro všechny strategie.

Speciálním případem her s konstantním součtem je hra s nulovým součtem, kde platí: co jeden hráč vyhraje, druhý prohraje (pokud hrají dva hráči).

1.3.5 Podle informací o důsledcích volby

- Hry s úplnou informací
- Hry s neúplnou informací

U her s úplnou informací mají všichni hráči stejné informace o hře jako všichni ostatní.

1.4 Maticové hry

Hru dvou hráčů s nulovým součtem a konečnými prostory strategií

$$S = \{s_1, s_2, \dots, s_m\}, T = \{t_1, t_2, \dots, t_n\}$$

lze zadat pomocí matice A ,

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} = \begin{pmatrix} u_1(s_1, t_1) & u_1(s_1, t_2) & \dots & u_1(s_1, t_n) \\ u_1(s_2, t_1) & u_1(s_2, t_2) & \dots & u_1(s_2, t_n) \\ \dots & \dots & \dots & \dots \\ u_1(s_k, t_1) & u_1(s_k, t_2) & \dots & u_1(s_k, t_n) \end{pmatrix} \quad (8)$$

jejíž prvky udávají hodnoty výplatní funkce prvního hráče (výplatní funkce druhého hráče má vždy opačnou hodnotu): prvek a_{ij} je roven hodnotě výplatní funkce prvního hráče, zvolil-li strategii s_i , a protivník zvolil strategii t_j . [7,13]

Při hledání řešení maticové hry se hledá takový prvek matice, který splňuje současně obě dvě tyto podmínky:

- Je nejmenší ze všech prvků na svém řádku,
- Je největší ze všech prvků ve svém sloupci.

Takovýto prvek se nazývá **sedlový bod**. Žádný hráč si změnou své strategie nemůže zvýšit svou výhru oproti sedlovému bodu.

Jestliže maticová hra má sedlový prvek, můžeme říct, že hra má řešení v ryzích strategiích. Pokud sedlový prvek nemá, má řešení ve smíšených strategiích. Smíšené strategie vyjadřují, s jakou pravděpodobností má hráč jednotlivou ryzí strategii zvolit.

1.5 Antagonistické hry

Základním modelem antagonistických her je maticová hra dvou hráčů v normálním tvaru s nulovým součtem.

Jedná se o hru, ve které vystupují dva inteligentní hráči, kteří se snaží maximalizovat svoji výhru, po volbě svých rozhodnutí se rozdělí o předem stanovenou částku. Tuto částku neovlivňuje volba jejich strategií. Matematickým modelem antagonistického konfliktu je hra v normálním tvaru s konstantním součtem [7]:

$$\{Q = \{1,2\}; S; T; u_1(s,t), u_2(s,t)\} \quad (4)$$

$$u_1(s,t) + u_2(s,t) = \textit{konst.} \textit{ pro každé } (s,t) \in S \times T$$

U antagonistického konfliktu se předpokládá, že zvýšení výhry u jednoho hráče vede automaticky ke snížení výhry hráče druhého. Strategie $\bar{s} \in S$ a $\bar{t} \in T$ nazveme *optimálními strategiemi*, pokud se vždy sníží výhra prvního hráče při volbě jiné strategie $s \in S$ a podobně pokud i druhý hráč si změnou své strategie $t \in T$ nemůže zvýšit svoji výhru.

Postup hledání optimální strategie je takový, že první hráč nejprve zvolí mezi svými strategiemi, neboli řádky matice (8), tu, která obsahuje prvek, jenž mu zaručuje nejvyšší výhru.

Druhý hráč postupuje podobně. Protože nejvyšší výhra prvního hráče znamená nejnižší výhru druhého hráče, vybere takovou strategii, tj. sloupec matice (8), kde je prvek, který mu zaručuje nejmenší maximální prohru.

		Hráč 2				
		t_1	t_2	t_3	t_4	
Hráč 1	s_1	5	4	4	5	4
	s_2	-4	5	3	9	-4
	s_k	7	8	-1	8	-1
max:		7	8	4	9	min

Obrázek 3: Příklad optimální strategie v maticové hře [7]

Dvojice strategií (s_1, t_3) je rovnovážným bodem hry. Můžeme ji tedy matematicky zapsat:

$$\max_s \min_t u_1(s_i, t_j) = 4 = \min_t \max_s u_1(s_i, t_j) = u_1(s_1, t_3)$$

2 POUŽITÉ TECHNOLOGIE

2.1 Programovací jazyk

Aplikace byla napsána pomocí programovacího jazyka PHP. PHP je netypový jazyk, který pracuje na straně serveru, je velmi úzce spojen s HTML, od verze 5 podporuje objektové programování. Je zpracováván na straně serveru pomocí interpretu PHP. Velmi často se využívá ve spojení s některým z databázových systémů, např. MySQL, MS SQL, SQLite. Syntaxe vychází z jazyků C, Java a Perl.

PHP kód se vkládá do HTML kódu pomocí párových značek, které značí začátek a konec kódu určeného ke zpracování PHP interpretem. Značky mají různou syntaxi a v přehledu jsou zvýrazněny tučně:

- `<? php_kód ?>`
- `<?php php_kód ?>`

V PHP je napsáno a vyvíjeno několik frameworků, které obsahují mnoho knihoven pro práci. K těm největším patří např. Zend.

2.2 Nette Framework

Frameworky jsou knihovny, které mají sloužit k tomu, aby ulehčily práci při programování, tzn. psaní méně kódu, více přehlednějšího, a hlavně rychlejší vývoj aplikace. Frameworky poskytují mnoho knihoven (tříd) pro základní prvky používané ve většině webových aplikací, je to zejména databázový layout pro snadnější vytváření SQL dotazů a komunikaci s databází (databázový layout NotORM je podrobněji popsán v Kapitole 2.5), knihovnu pro přihlašování uživatelů, stránkování výstupních dat v prohlížeči, použití AJAXu, zasílání e-mailů atd.

V případě webových frameworků se často zmiňuje MVC, což je architektura rozdělující aplikaci na nezávislé vrstvy. [9] Podrobněji se architektuře MVC věnuje Kapitola 2.4.

Nette Framework je napsaný v PHP 5 s plným využitím objektů (OOP). Jeho licence, která vychází z BSD, patří k těm nejvolnějším. Podle testu uveřejněnému na serveru Root.cz je jedním z nejvýkonnějších frameworků. [9]

2.3 MVC

Architektura MVC, neboli Model-View-Controller (v Nette se používá pro Controller pojmenování Presenter), rozděluje aplikaci na tři různé logické části tak, aby pracovali co nejvíce nezávisle na sobě, a zároveň přesně definuje, za co je která část v aplikaci zodpovědná.

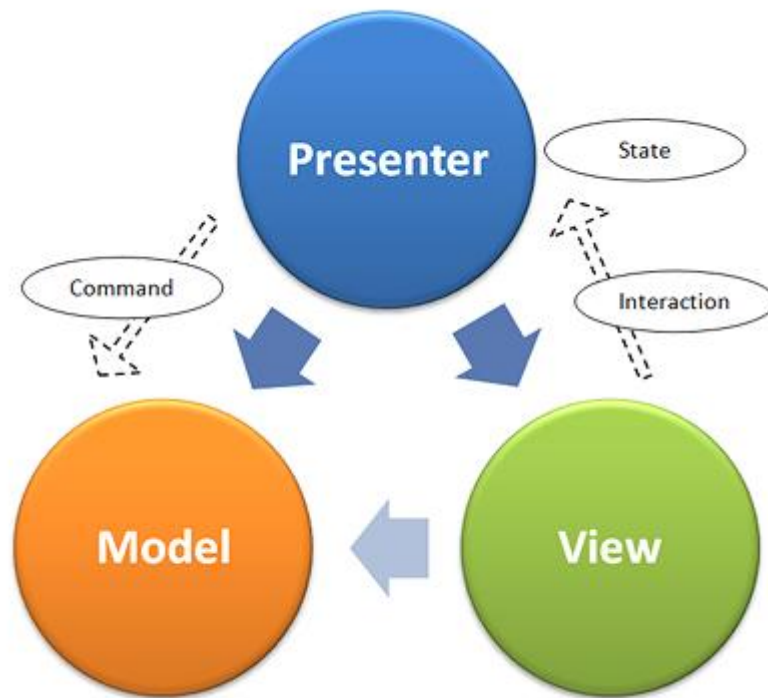
Model je funkčním a datovým základem celé aplikace. Nabízí rozhraní pro přístup k datům, která mohou být uložena např. v databázi. Nejčastěji je implementován pomocí tříd – jeden model = jedna třída. V přístupech odlišných od OOP jej lze realizovat i pomocí funkcí.

View poskytuje uživateli náhled na model a jeho stav, a nějakým způsobem ho prezentuje. Při změně stavu modelu může dojít ke změně *View*.

Controller (Presenter) definuje chování aplikace. Zpracovává veškeré uživatelské vstupy a události. Po zpracování může pracovat s modelem a na základě výsledku akce určuje vhodné *View*.

Podle [11] mezi hlavní výhody MVC patří:

- Snadné zpřístupnění pro různé druhy klientů (např. prohlížeče mobilních telefonů). Stačí pouze vytvořit nové *View*, *Model* zůstává stejný.
- Minimalizace duplicitního kódu, například bez odděleného a zapouzdřeného *Modelu* by se pro každé nové *View* musela znovu programovat aplikační logika.
- Rozdělení vývojářských rolí, jednotlivé části pak mohou být vyvíjeny nezávisle na sobě, stačí mít nadefinované jednotlivá rozhraní.
- Znovupoužitelnost kódu.
- Snadná rozšiřitelnost aplikace - modularita.
- V případě použití nadřazeného *BasePresenteru*, který zpracovává všechny přicházející požadavky, lze jednotně kontrolovat přístupová práva nebo kontrolu autentifikace uživatele.



Obrázek 4: Architektura Model-View- Presenter [9]

Na Obrázku 5 jsou znázorněny vazby mezi jednotlivými prvky architektury MVC. V principu existují pouze dvě přímé vazby:

- Presenter má přímý odkaz na Model (úprava dat jeho dat)
- View má přímý odkaz na Model, aby mohl jeho data zobrazit

Žádné další vazby nejsou na této úrovni podstatné, ačkoliv v praxi a v závislosti na konkrétní variaci MVC je ještě poměrně častá vazba mezi Presenterem a View. [9]

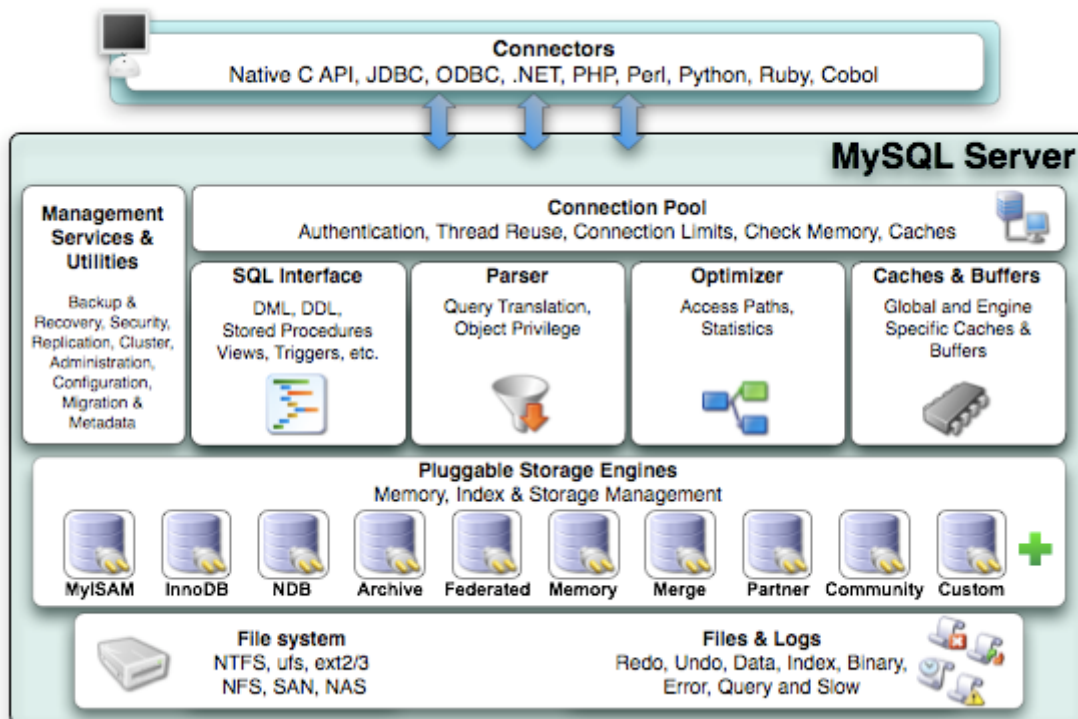
Naopak přímá vazba Modelu na ostatní dvě komponenty nesmí existovat, to by byla chyba v návrhu aplikace.

2.4 Databázový systém

Pro tuto aplikaci byl zvolen relační databázový systém MySQL, jeden z nejrozšířenějších, je vyvíjen firmou Oracle pod licencí GNU GPL. Výhody MySQL jsou podle [8] zejména :

- Stabilita a rychlost
- Podpora multi-users a multi-threads

- Umožňuje přístup z mnoha jazyků (C, C++, Perl, PHP aj.)
- Pracuje ve většině operačních systémů



Obrázek 5: Architektura MySQL [8]

MySQL obsahuje několik typů úložišť dat, mezi ty nejpoužívanější patří InnoDB a MyISAM.

2.4.1 MyISAM

MyISAM bylo dlouhou dobu základním typem úložiště pro MySQL, od verze MySQL 5.5 ho však nahradilo InnoDB [8]. Jedním z důvodů byla absence podpory referenční integrity dat a transakcí. MyISAM podporuje zamykání na úrovni tabulek (table level locking) a fulltextové vyhledávání.

Pro každou tabulku v databázi jsou vytvořeny na disku tři soubory, první nese definici tabulky, druhý obsahuje data a ve třetím jsou uloženy indexy. [8]

2.4.2 InnoDB

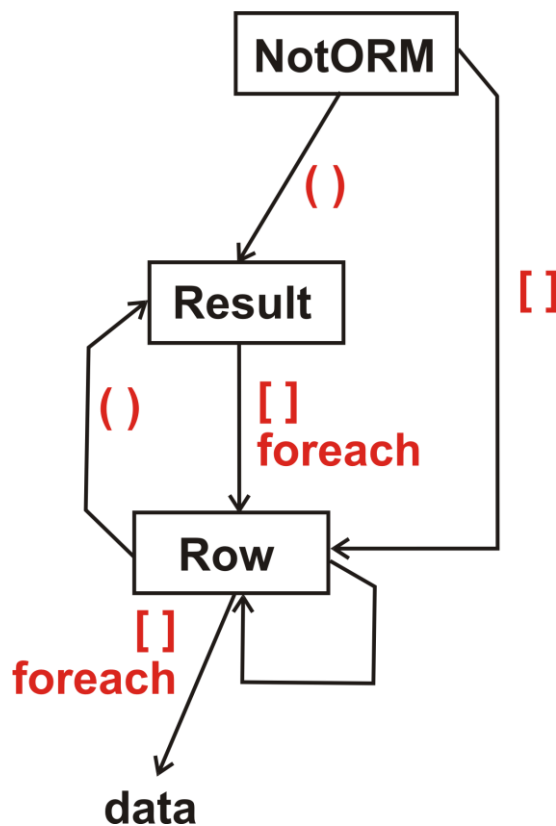
InnoDB je úložiště, které podporuje transakce, cache paměť pro data a indexy, zachovává referenční integritu dat pomocí cizích klíčů a podporuje zamykání už na úrovni řádků (row

level locking). Obsahuje také funkci automatického obnovení ze zálohy při ztrátě dat. Tabulky se ukládají na disku do tzv. tablespace. [8]

Nevýhodou je, že nepodporuje fulltextové vyhledávání.

2.5 NotORM

NotORM je knihovna napsaná v jazyce PHP pro práci s daty v databázi, pracuje podobně jako PHP knihovna SimpleXML s XML dokumenty. Vyžaduje PHP od verze 5.1 a databázi s podporou PDO.



Obrázek 6: Schéma fungování NotORM [9]

NotORM redukuje složité a dlouhé dotazy do minimálního počtu jednoduchých krátkých dotazů a z databáze přenáší jen ta data, která jsou potřeba. Dosahuje se toho dvěma způsoby:

- Získání souvisejících záznamů se provádí najednou pro všechny řádky výsledku jedním dotazem.

- Přenášejí se jen ty sloupce, které se nakonec skutečně použijí (při povolení této vlastnosti).

Knihovna nepoužívá cache výsledků dotazů, protože její filozofií je to, že taková cache zastarává a zbytečně zabírá paměť, která by mohla být využita k něčemu užitečnějšímu, a je tedy lepší získávat data efektivně přímo z databáze.

2.6 jQuery

jQuery je rozsáhlý JavaScriptový framework, který zjednodušuje práci z DOM elementy, událostmi, animacemi a AJAXem. Jelikož většina prvků je zpracovávána JavaScriptem v prohlížeči, přináší do webové aplikace interaktivitu.

2.6.1 jQuery PLUGINS

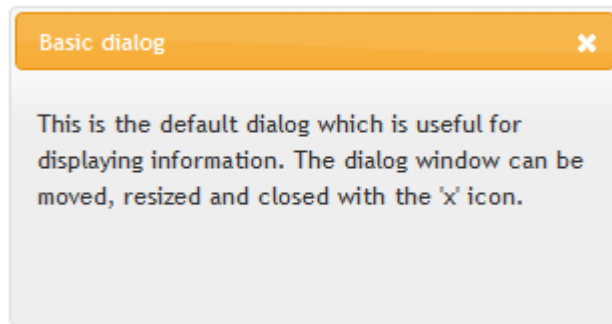
Mnoho programátorů vyvíjí s pomocí jQuery vlastní doplňky pro web a nabízí je volně ke stažení. Mezi populární patří např. stromové menu, obrázkové slideshow, live validace formulářů atd.

2.6.2 jQuery UI

jQuery UI poskytuje různé grafické prvky jako efekty a funkční prvky. Tyto prvky mohou sloužit ke sběru vstupních dat nebo k zobrazení výstupu. Na Obrázku 7 je prvek kalendáře pro výběr data, na Obrázku 8 jednoduché dialogové okno.



Obrázek 7: jQuery – Datepicker



Obrázek 8: jQuery – Dialog

3 HRY V INTERNETOVÉM PROHLÍŽEČI

Hry pro internetové prohlížeče, tzv. online hry, jsou hry, pro jejichž hraní je potřeba mít připojení k internetu a internetový prohlížeč (např. Microsoft Internet Explorer, Mozilla Firefox, Google Chrome). Velkou výhodou těchto her je, že nepotřebují mít nainstalovaný pro svůj běh žádný dodatečný software. Nevýhodou bývá většinou nižší grafická úroveň a omezenější funkcionalita. Dle použité technologie se dělí na dvě skupiny.

3.1 Hry na straně serveru

Tyto aplikace jsou většinou programovány v jazycích PHP, ASP/ASP.NET, PERL nebo Ruby, které obstarávají veškerou logiku, zpracovávají uživatelské vstupy ve formě formulářů či akcí a vracejí dynamicky generovaný obsah, většinou ve formě HTML. Pro ukládání dat o hráčích a herních stavech se využívá databáze. Jednou z nejrozšířenějších je databáze MySQL šířená pod licencí GNU GPL. Z dalších byl zmínil ještě PostgreSQL, MS SQL či jednoduchý SQLite.

Ke psaní aplikací je výhodné, ne-li žádoucí, využít frameworky (Nette, Zend, Ruby on Rails), které jsou plně objektové a obsahují knihovny pro tvorbu a zpracování formulářů, uživatelských práv atd. Obsahují také většinou databázový layout zjednodušující tvorbu SQL dotazů a rutin, se kterými se programátor běžně setkává (např. Dibi nebo NotORM) a mají také zabudovanou ochranu proti útokům typu SQL Injection.

Tato technologie je zejména vhodná pro tahové hry, kdy hru hraje více hráčů a neočekává se okamžitá reakce protihráče na vzniklou situaci.

3.2 Hry na straně klienta

Pro tento typ her je využito např. technologie Adobe Flash, pro jejichž spuštění je nutné, aby v prohlížeči byl nainstalován Adobe Flash Player. Adobe Flash je grafický vektorový program, který má implementovaný svůj objektově orientovaný programovací jazyk Action Script, pomocí kterého lze tvořit animace i logiku aplikace. Samotný program je potom vložen do HTML jako objekt.

Další technologie, která přináší interaktivitu do prohlížeče, je JavaApplet. Ten má vymezený určitý prostor obrazovky, ze kterého může číst a zapisovat do něj. O interpretaci kódu se stará JVM. Tato technologie je multiplatformní.

Ještě je třeba zmínit ActiveX, což je binární komponenta vkládaná do stránky podobně jako JavaApplet. Funguje ale pouze na 32-bitových operačních systémech Windows s prohlížečem Microsoft Explorer. Vytvořit ji lze v libovolném programovacím jazyce (Visual C++, Delphi atd).

II. PRAKTICKÁ ČÁST

4 HERNÍ MECHANISMY HRY MĚSTO ZLODĚJŮ

Hra, která byla v rámci této práce vytvořena, dostala název Město zlodějů. Registrací získá hráč jednu postavu, se kterou může procházet město a využívat jeho možností.

4.1 Popis hry

Cílem hry je stát se nejbohatší postavou ve městě, tzn. mít co nejvíce peněz - Buggů.

Každý hráč si může zdokonalovat obranné a útočné vlastnosti své postavy koupí zbraní v obchodě se zbraněmi. K tomu je třeba mít peníze, které se dají získat buď výhrou v casinu, nebo na ulici oloupením jiného hráče. Navíc každý hráč dostává stejný týdenní plat, částka je mu připsána na účet každých sedm dní. Tato částka činí 500 Buggs.

V casinu, kde se hraje hra GOLDtip, může hráč hodně vyhrát, ale taky hodně ztratit. Záleží na jeho zvolené strategii. Vyhrané peníze může opět použít k další hře nebo je jít uložit do bezpečí, do banky. Ovšem každým přesunem po ulici dává hráč příležitost ostatním hráčům k tomu, aby jej okradli, protože zloději můžou krást právě tam. Naštěstí existují i místa, kde jsou hráči před okradením chráněni a to vždy uvnitř budov.

Na ulicích tedy není bezpečno a nosit u sebe hotovost se nemusí zrovna vyplatit. Proto mají hráči možnost si za své vydělané peníze nakoupit zbraně. Prodejna zbraní je úplně až na severu města, a cesta k ní netrvá zrovna chvíli. Zbraně nejsou zrovna levné a je třeba je platit v hotovosti. Záleží pak na hráči, jak s koupenou zbraní naloží. Nic mu nebrání v tom, aby ji použil k přepadávání ostatních hráčů. Zloději ovšem musí počítat s tím, že po přepadení nějaký čas zůstávají na místě, kde loupež provedli.

Záleží pouze na hráči, jakou cestou se vydá za svým ziskem a jak dosáhne předních míst ve společnosti.

4.2 Mapa města

Město zlodějů má několik důležitých budov, je to banka, casino a obchod se zbraněmi. Tyto budovy jsou od sebe poněkud vzdáleny a jediný způsob, jak se mezi nimi přesouvat, je jít pěšky po ulici. Pro lepší představu byl vytvořen plán města (Obrázek 9). Vzdálenost mezi nimi je znázorněna v Tabulce 1 a je uváděna v časových jednotkách.

	Banka	Casino	Obchod
Banka	-	30 min	50 min
Casino	30 min	-	20 min
Obchod	50 min	20 min	-

Tabulka 1: Vzdálenosti mezi budovami



Obrázek 9: Plán města

4.3 Začátek hry

Při prvním přihlášení začíná hráč v bance, kde má na účtu malou částku do začátku hry. Jeho postava nemá u sebe žádnou zbraň, to znamená, že obranná i útočná síla jsou na úrovni nula.

4.4 Banka

Banka se nachází ve středu města. Každý hráč u ní má vedený účet a může kdykoliv, když je v bance, provádět operace se svým účtem (vybírat nebo vkládat peníze). Tyto transakce nejsou nijak zpoplatněny, ani samotné vedení účtu. Co banka neumožňuje, je posílat peníze mezi účty dvou hráčů. Banka také neposkytuje půjčky a slouží tedy jako bezplatná úschovna peněz před zloději.

Banka je bezpečné místo, kde nedochází k přepadením.

4.5 Casino

Casino se nachází v severní části města. V casinu lze hrát pouze jeden typ hry s názvem GOLDtip (Kapitola 6).

Hráč se může do hry zapojit dvěma způsoby. Buď ji může vytvořit a čekat, než se připojí jiný hráč, nebo může procházet hry vytvořené jinými hráči a k některé z nich se připojit. Je dovoleno účastnit se současně pouze jedné hry, takže hráč nemůže mít vytvořenou svou hru a zároveň se připojit ke hře jiného hráče. Pokud hráč hraje hru, nesmí samozřejmě opustit kasino.

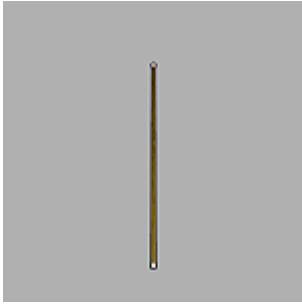
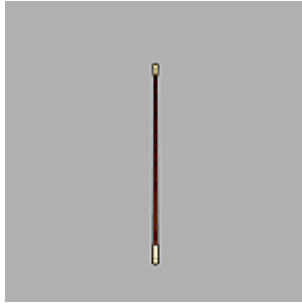

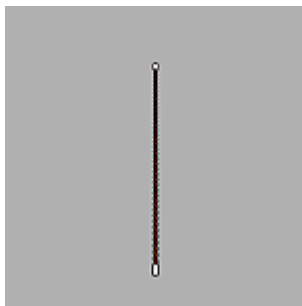
Casino také dovoluje hráčům nahlížet do historie odehraných her a tím dává příležitost vypořádat nejpravděpodobnější strategii protihráčů.

Casino je stejně jako banka bezpečné místo, kde nedochází k přepadením.





4.6 Obchod se zbraněmi

Obchod se nachází na severozápadě, v okrajové části města. Lze zde zakoupit mnoho obranných i útočných zbraní. Platí se hotově, takže k nákupu je třeba mít potřebný obnos u sebe. V obchodě lze také zbraň prodat, ovšem za zlomek její nákupní ceny. V aplikaci byla nastavena výkupní cena na 50 % z ceny původní.





4.6.1 Přehled zbraní

	<p>Dřevěná hůl</p> <p>290 Buggů +0 Obrana +1 Útok</p>		<p>Dřevěná hůl s železnými konci</p> <p>600 Buggů +0 Obrana +2 Útok</p>
	<p>Dřevěná vyvážená hůl</p> <p>700 Buggů +1 Obrana +2 Útok</p>		<p>Železná kovaná hůl</p> <p>800 Buggů +2 Obrana +2 Útok</p>


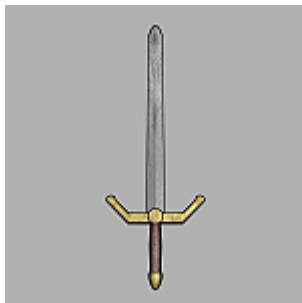


Tabulka 2: Přehled druhů holí

	<p>Dubový prak</p> <p>1600 Buggů +2 Obrana +4 Útok</p>		<p>Dalekonosný prak</p> <p>1900 Buggů +3 Obrana +4 Útok</p>
	<p>Přesný kalibrováný prak</p> <p>2300 Buggů +4 Obrana +4 Útok</p>		<p>Zesílený obranný prak</p> <p>2900 Buggů +5 Obrana +4 Útok</p>





Tabulka 3: Přehled druhů praků

	<p>Kuchyňský nůž</p> <p>3500 Buggů +3 Obrana +6 Útok</p>		<p>Dýka</p> <p>3900 Buggů +4 Obrana +6 Útok</p>
	<p>Vrhací dýka</p> <p>4400 Buggů +5 Obrana +6 Útok</p>		<p>Dýka se záštitou</p> <p>4900 Buggů +6 Obrana +6 Útok</p>





Tabulka 4: Přehled druhů dýk

	<p>Dřevěný středně dlouhý meč</p> <p>5200 Buggů +6 Obrana +8 Útok</p>		<p>Dlouhý meč</p> <p>6500 Buggů +7 Obrana +8 Útok</p>
	<p>Krátký jednoučinný meč</p> <p>7800 Buggů +8 Obrana +8 Útok</p>		<p>Obouruký vyvážený meč</p> <p>8500 Buggů +9 Obrana +8 Útok</p>





Tabulka 5: Přehled druhů mečů

	<p>Dřevorubecká sekera</p> <p>9000 Buggů +8 Obrana +10 Útok</p>		<p>Jezdecká sekera s bodcem</p> <p>9800 Buggů +9 Obrana +10 Útok</p>
	<p>Dvousečná sekera</p> <p>11000 Buggů +10 Obrana +10 Útok</p>		<p>Sekera s hrotem</p> <p>12000 Buggů +11 Obrana +10 Útok</p>





Tabulka 6: Přehled seker

	<p>Železný tupý palcát</p> <p>15000 Buggů +10 Obrana +12 Útok</p>		<p>Obyčejný palcát</p> <p>18000 Buggů +11 Obrana +12 Útok</p>
	<p>Palcát s ostřím</p> <p>21000 Buggů +12 Obrana +12 Útok</p>		<p>Odlehčený palcát s ostřím</p> <p>25000 Buggů +13 Obrana +12 Útok</p>

Tabulka 7: Přehled druhů řemdihů

	<p>Řemdih s tupou hlavicí</p> <p>27000 Buggů +12 Obrana +14 Útok</p>		<p>Řemdih s ostrou údernou hlavicí</p> <p>30000 Buggů +13 Obrana +14 Útok</p>
	<p>Řemdih se dvěma hlavicemi</p> <p>32000 Buggů +14 Obrana +14 Útok</p>		<p>Řemdih se třemi hlavicemi</p> <p>35000 Buggů +15 Obrana +14 Útok</p>

Tabulka 8: Přehled druhů palcůtů

	<p>Vikingský štít</p> <p>500 Buggů +1 Obrana +0 Útok</p>		<p>Pěší lehký štít</p> <p>600 Buggů +2 Obrana +0 Útok</p>
	<p>Jezdecký štít</p> <p>1400 Buggů +4 Obrana +0 Útok</p>		<p>Gotický kovový štít</p> <p>2000 Buggů +5 Obrana +0 Útok</p>

Tabulka 9: Přehled druhů štítů

	<p>Kožená lehká zbroj</p> <p>4000 Buggů +6 Obrana +0 Útok</p>		<p>Kroužková zbroj</p> <p>7000 Buggů +8 Obrana +0 Útok</p>
	<p>Plechová zbroj</p> <p>10000 Buggů +10 Obrana +0 Útok</p>		<p>Plátovaná zbroj</p> <p>15000 Buggů +12 Obrana +0 Útok</p>

Tabulka 10: Přehled druhů zbroje

Všechny grafické prvky uvedené v této kapitole vymodeloval v rámci své bakalářské práce student Zbyněk Máčala v programu Blender.

4.7 Postava

Každému uživatelskému účtu je přidělena právě jedna postava. Při registraci si hráč zvolí její jméno a obrázek. Tento obrázek není možné později změnit, protože jednoznačně identifikuje hráče v průběhu hry. Každá postava může u sebe nosit libovolnou hotovost a nejvýše jednu zbraň.

4.7.1 Útočné vlastnosti

Postava má útočnou sílu, která je určena zbraní, kterou má u sebe. Pokud nemá u sebe žádnou zbraň, její útočná síla je rovna nule. Zakoupit zbraň lze pouze v obchodě se zbraněmi (Kapitola 5.3).

4.7.2 Obranné vlastnosti

Obranná síla slouží na obranu při přepadení jiným hráčem. Opět platí stejné pravidlo jako výše – sílu určuje zbraň v rukou postavy.

4.8 Ulice

Ulice spojuje jednotlivé budovy, a pokud chce hráč jít z jedné budovy do jiné, tak musí jít po ní, čímž dává příležitost ostatním hráčům, aby ho okradli o hotovost, kterou má u sebe. Ovšem pobyt na ulici dává i jemu příležitost někoho přepadnout.

4.8.1 Přepadení

Hráč na ulici vidí kolem sebe ostatní hráče, ale pouze na určitou vzdálenost. Nevidí však, jak jsou postavy silné, ani kolik mají u sebe hotovosti. Přepadnout může pouze toho, koho vidí. Zloděj musí počítat s tím, že po přepadení setrvává na místě činu určitý čas, po který nemůže nic dělat. A to i v případě neúspěšného přepadení.

Přepadení je vyhodnoceno jako úspěšné tehdy, je-li splněna podmínka:

$$\text{Útočné vlastnosti útočníka} > \text{Obranné vlastnosti přepadeného} \quad (1)$$

O útočných a obranných vlastnostech postavy pojednává Kapitola 4.7.1 a 4.7.2.

V Tabulce 11 je uveden příklad vlastností dvou postav A a B. Pokud postava A přepadne postavu B, je splněna podmínka (1), útok je úspěšný a útočnickovi je připsána veškerá hotovost postavy A. Pokud by ovšem postava B přepadla postavu A, podmínka (1) splněna není a útok je vyhodnocen jako neúspěšný. V obou případech musí zloději zůstat na místě činu po určitý čas, neschopni jakékoliv akce.

	Útočné vlastnosti	Obranné vlastnosti	Hotovost
Postava A	8	6	250 Buggů
Postava B	6	3	1200 Buggů

Tabulka 11: Příklad přepadení

5 HRA GOLDTIP

Jedná se o konečnou maticovou hru dvou hráčů s nulovým součtem, čili antagonistickou hru. Kde výhra jednoho hráče je rovna prohře druhého hráče.

Hra je určena pouze pro dva hráče. Zakladatel hry zvolí základní vklad a tuto částku vloží do společného banku. Protihráč musí pro vstup do hry vložit do banku stejnou částku. Z banku je vyčleněno 20 % pro vítěze, tzv. bonus, a zbytek je náhodně rozdělen na tři hromádky.

Každý hráč má k dispozici celkem sto kreditů, které rozdělí k jednotlivým hromádkám dle svého uvážení. Na toto rozhodnutí má určitý čas. Až tak učiní oba hráči, tak se vyhodnotí počty kreditů u hromádek. Hromádka připadne tomu hráči, který u ní má více kreditů. Pokud se stane, že u hromádky mají stejně, hromádku si rozdělí napůl. Navíc kdo vyhrál více hromádek, tomu připadne ještě bonus.

5.1.1 Partie hry v normálním tvaru

Jelikož se velikost hromádek při každé partii mění, nelze tak zaznamenat obecně hru GOLDtip v normálním tvaru. Proto byl model uveden pro jednu konkrétní partii. Velikost hromádek byl zvolen stejný jako v příkladu (Kapitola 5.1.2). Pro názornější a přehlednější zobrazení byly kredity každého hráče omezeny pouze na tři.

Jednotlivé strategie byly označeny písmeny s a t . Kladná čísla udávají velikost výhry hráče B, záporná hráče A, neboť se jedná o antagonistickou hru s nulovým součtem.

Sedlový bod této matice neexistuje pro ryzí strategie. Optimální řešení ale přesto existuje ve smíšených strategiích, a to hrát všechny ryzí strategie s určitou pravděpodobností. Pokud by některý hráč hrál stále jednu strategii, inteligentní protihráč toto jistě zaznamená a v další hře toho využije ke své výhře.

		HRÁČ B								
		t1	t2	t3	t4	t5	t6	t7	t8	t9
		(0,0,3)	(0,3,0)	(3,0,0)	(0,1,2)	(0,2,1)	(1,0,2)	(2,0,1)	(1,2,0)	(2,1,0)
HRÁČ A	s1 (0,0,3)	0	-160	-198	-160	-160	-198	-198	228	228
	s2 (0,3,0)	160	0	-38	160	160	548	548	-38	-38
	s3 (3,0,0)	198	38	0	624	624	198	198	38	38
	s4 (0,1,2)	160	-160	-624	0	-160	-38	-624	228	-198
	s5 (0,2,1)	160	-160	-624	160	0	548	-38	-198	-624
	s6 (1,0,2)	198	-548	-198	38	-548	0	-198	-160	228
	s7 (2,0,1)	198	-548	-198	624	38	198	0	-548	-160
	s8 (1,2,0)	-228	38	-38	-228	198	160	548	0	-38
	s9 (2,1,0)	-228	38	-38	198	624	-228	160	38	0

Obrázek 10: Partie hry GOLDtip v normálním tvaru

5.1.2 Příklad hry

Do hry vstoupili dva hráči, A a B. Hru vytvořil hráč A s vyšší vkladu 1000 Buggs, tedy společný bank činí 2000 Buggů, z banku je vyčleněn bonus pro vítěze 400 Buggs (20 %) a zbylých 1600 Buggs je rozděleno náhodně na tři hromádky: 376 Buggs, 452 Buggs a 772 Buggs. Potom co každý hráč rozdělí svých sto kreditů k jednotlivým hromádkám se hra vyhodnotí. V tabulce je znázorněn průběh hry a výhry jednotlivých hráčů.

	Hráč A	Hráč B	Výhra hráče A	Výhra hráče B
376 Buggs Hromádka 1	0 kreditů	40 kreditů	0 Buggs	376 Bugg
452 Buggs Hromádka 2	20 kreditů	50 kreditů	0 Bugg	452 Buggs
772 Buggs Hromádka 3	80 kreditů	10 kreditů	772 Buggs	0 Bugg
400 Buggs Bonus			0	400 Buggů
2000 Buggs Společný bank			772 Buggs	1228 Buggs

Tabulka 12: Příklad hry Goldtip se společným bankem 2000 Buggů

5.1.3 Vytvoření hry

Při vytváření hry se nejprve zvolí výše vkladu, která půjde do společného banku, potom je hra vytvořena a zveřejněna ostatním hráčům v casinu.

Autor hry může hru zrušit do chvíle, než se připojí jiný hráč.

5.1.4 Připojení ke hře

Pro připojení k vytvořené hře je třeba mít u sebe potřebnou hotovost na složení vkladu. Po připojení hráč vidí jednotlivé hromádky a může k nim přidělit kredity. Má na to však omezený čas.

5.1.5 Hra proti počítači

Hra proti počítači je ve formě zkušební hry. Hráč navolí výši vkladu, rozdělí svých sto kreditů ke hromádkám, při vyhodnocení hry počítač rozdělí též svých sto kreditů.

Počítač hraje optimální strategii, proto sto kreditů náhodně rozdělí na tři části. K největší hromádce prisoudí nejvíce kreditů a k nejmenší hromádce přidělí nejméně kreditů.

Hráči je poté zobrazen výsledek hry a jeho případný zisk.

GOLDtip

[Má hra](#) / [Odehrané hry](#) / [Zkušební hra](#) / [Pravidla](#)

Hra

Vytvořena: 04.06.2011 10:14:34
Stav: Čeká se na soupeře.
Akce: [Zrušit hru](#)

Vklad kadého hráče: 1000 Buggs
Bonus pro vítěze: 400 Buggs

376 Buggs 452 Buggs 772 Buggs

0 0 0

100
zbývající kredit

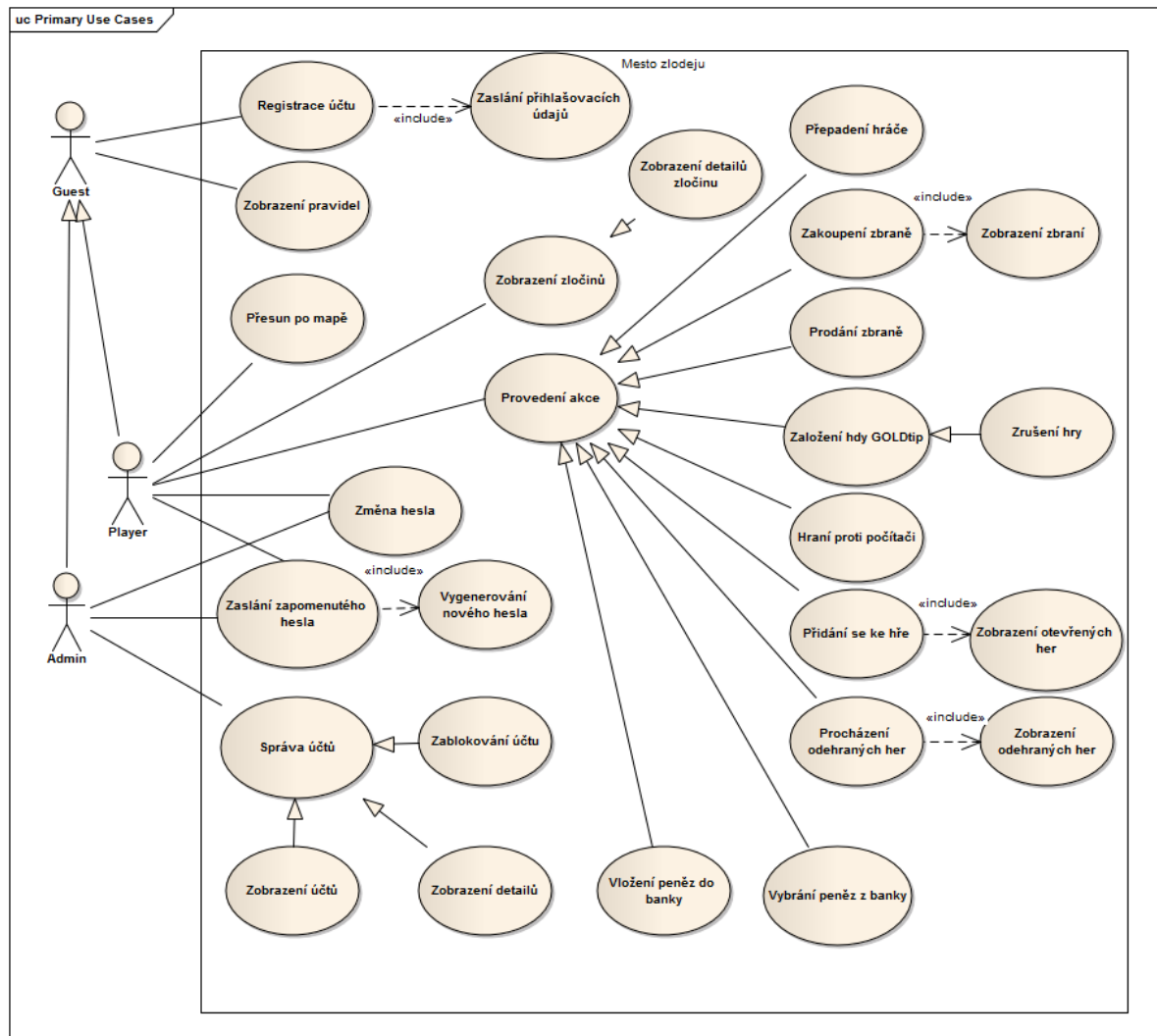
[Uložit tipy](#)

The screenshot shows a game interface for 'GOLDtip'. At the top, there are navigation links: 'Má hra', 'Odehrané hry', 'Zkušební hra', and 'Pravidla'. Below this is the title 'Hra'. The game status is 'Čeká se na soupeře.' (Waiting for opponent). There are three player bug counts displayed: 376, 452, and 772. Below each count is a vertical bar with a '0' at the top, indicating the current bet for each player. At the bottom, there is a green bar representing the remaining credit, with the value '100' and the text 'zbývající kredit' (remaining credit). A button labeled 'Uložit tipy' (Save bets) is located at the bottom left.

Obrázek 11: GOLDtip – ukázka partie

6 APLIKACE

6.1 Případy užití



Obrázek 12: Případy užití

6.1.1 Registrace účtu

Registrace účtu obnáší získat od uživatele údaje, na základě kterých je vytvořen účet k přihlášení do hry, k účtu je také přiřazena postava, se kterou hráč bude hrát.

Jako přihlašovací údaje jsou použity e-mailová adresa v kombinaci s heslem, které si uživatel zvolí. Všechny formuláře prochází validací pomocí JavaScriptu na straně klienta a PHP na straně serveru. O provedení validace se stará třída *AppFrom* z frameworku Nette. Při vytváření účtu je kontrolováno, zdali již existuje v databázi účet se zadaným e-mailem.

Tato validace musí proběhnout na straně serveru, neboť je potřeba se dotazovat do databáze. Pokud již e-mail existuje, registrace se neprovede a je třeba zadat jiný. Na obrázku 12 je ukázka kódu, který tuto validaci provádí. Vstupním parametrem funkce je prvek formuláře, což je objekt. Jeho metoda `getValue()` vrací obsah prvku. V případě existence e-mailu v databázi vrátí validační funkce hodnotu `FALSE`.

```
173  /**
174      * Check if the user already exists in DB.
175      * @param FormItem
176      * @return boolean
177      */
178  public static function exist($item) {
179      $numbers = count(
180          Model::$database->users('id')
181          ->where('login = ?', $item->getValue())
182      );
183      return $numbers == 0;
184  }
```

Obrázek 13: Funkce pro kontrolu existence účtu

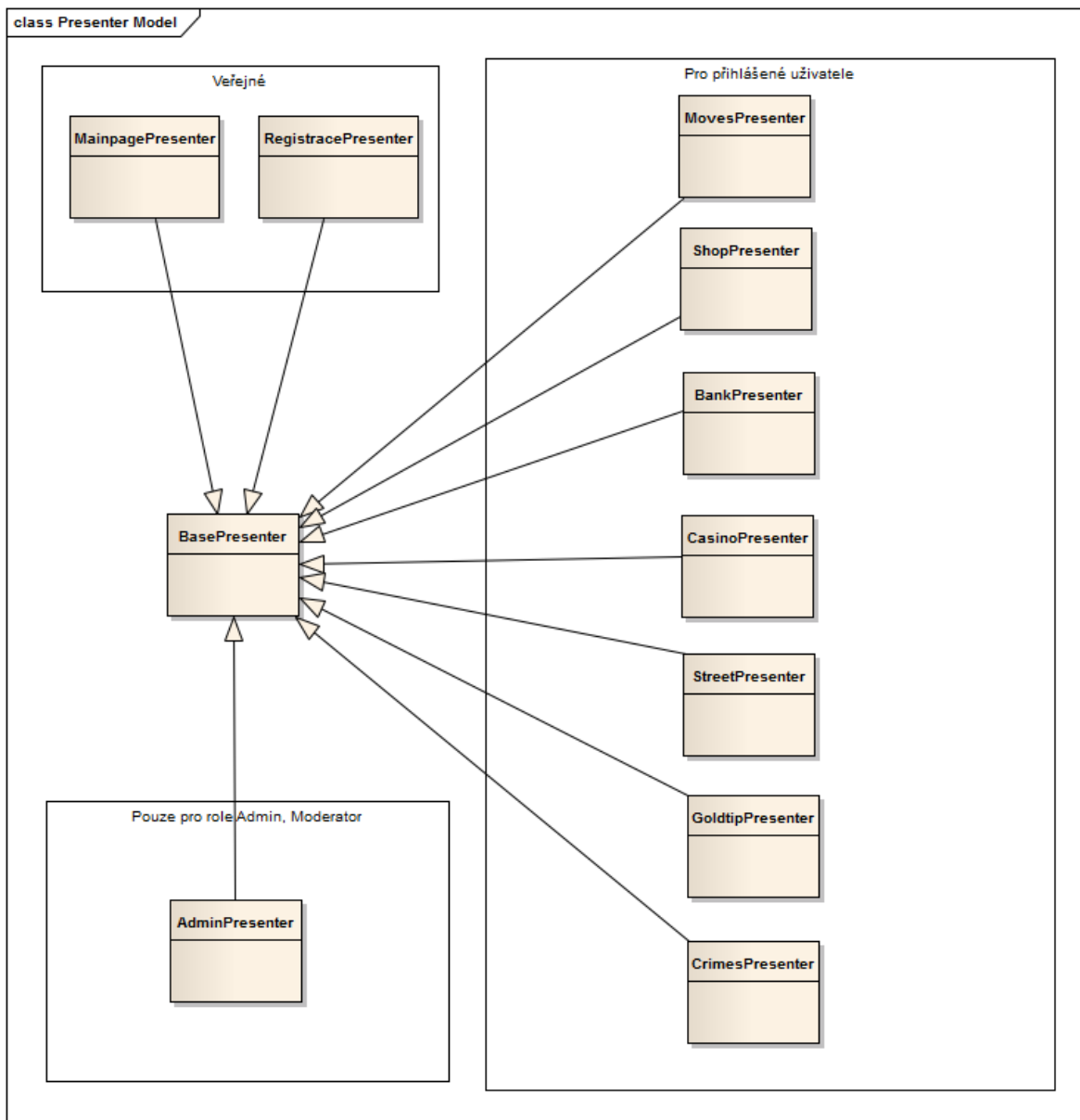
Při registraci si hráč může nahrát obrázek své postavy, pro nahrání obrázku je ve formuláři využito AJAXu. Formulář obsahuje atribut `target`, který směřuje odeslání formuláře do objektu `iframe`, čímž je zamezeno tomu, aby prohlížeč znovu načtl URL, ze které je formulář odeslán. Aplikace zpracuje nahraný obrázek, uloží jej do požadovaného adresáře, případně upraví jeho rozměry. Jakmile je nahrávání na server dokončeno, objekt `iframe` je znovu načten a tuto událost zachycuje JavaScript, který poté od aplikace získá adresu obrázku. Ta mu je předána ve formátu JSON: `{"element":{"img":"default.png"}}`.

Funkce pro získání nahraného obrázku postavy při registraci se volá: `registrace/get-image`. Pokud žádný obrázek nahrán nebyl, je nastaven `default.png`.

6.2 E-R model databáze

Návrh databáze je uveden v Příloze I.

6.3 Rozdělení na presentery



Obrázek 14: Rozdělení na presentery

Všechny presentery jsou potomkem BasePresenteru, ve kterém jsou implementovány společné funkce a zpracování důležitých událostí, které mohou nastat ve všech ostatních presenterech - přihlášení a odhlášení uživatele.

6.3.1 BasePresenter

```
72 public function signInFormSubmitted($form) {  
73     try {  
74         $this->user->login($form['login']->value,$form['password']->value);  
75         $this->user->setExpiration('45 minutes');  
76  
77         Logs::writeLog($this->user->getId());  
78  
79         $this->application->restoreRequest($this->backlink);  
80         $this->redirect('Mainpage:');  
81     } catch (NAuthenticationException $e) {  
82         $this->flashMessage($e->getMessage());  
83     }  
84 }
```

Obrázek 15: Metoda pro zpracování formuláře přihlášení uživatele

Zdrojový kód zpracování události přihlášení uživatele je na Obrázku 15. Na řádce 74 se předají parametry (login a heslo) objektu *\$user*, parametry jsou zkontrolovány s údaji v databázi a na základě takto ověřených informací načteny data o uživateli a jsou mu přiřazeny role. Pokud údaje neodpovídají s údaji v databázi, je vygenerována výjimka, která přeruší aktuální vykonávání kódu v sekci *try { }*, tato výjimka je zachycena a vypsána uživateli v podobě zprávy.

Na řádce 75 je nastavena expirace na 45 minut, pokud je uživatel po tuto dobu neaktivní, je automaticky odhlášen.

Zápis logů do databáze je prováděn na řádce 77, na řádce 79 se provádí načtení předchozího stavu aplikace, pokud byl uživatel odhlášen automaticky po uplynutí doby nečinnosti. Na řádce 80 je přesměrování, které je důležité z toho důvodu, aby se při obnovení stránky neodeslal formulář znovu.

BasePresenter taky obsahuje obsluhu události odhlášení uživatele, funkce pro zjišťování oprávnění a kontrolu platnosti session.

6.3.2 MainpagePresenter

Obstarává prezentaci dat, která jsou přístupná všem uživatelům, i nepřihlášeným. V aplikaci se jedná o zobrazení informací o hře a pravidel hry.

Mainpage/default – úvodní stránka.

Mainpage/pravidla – pravidla hry.

6.3.3 RegistracePresenter

Vytváří formulář pro registraci uživatele a zpracovává odeslaná data. Vytváří formulář pro zaslání zapomenutého hesla k účtu na zadaný e-mail.

Registrace/get-image – vrací obrázek pro registraci ve formátu JSON.

Registrace/default – formulář pro registraci nového účtu.

Registrace/zaslani – formulář pro zaslání zapomenutého hesla.

6.3.4 AdminPresenter

Uživateli s rolí *Admin* umožňuje procházet seznam účtů a nahlížet do detailů o účtech a postavách ve hře. Ostatním rolím umožňuje pouze přístup ke změně hesla.

Admin/default – seznam vytvořených účtů.

Admin/detail/<id> - detaily hráče s ID = <id>.

Admin/block/<id> - zablokování účtu hráče s ID = <id>.

Admin/upravit – formulář pro změnu hesla k účtu.

6.3.5 MovesPresenter

Obstarává nabídku míst, kam se může postava přesunout a zpracovává události přesunu.

Moves/default – zobrazení seznamu míst, kam se hráč může přesunout.

6.3.6 BankPresenter

Poskytuje operace s účtem hráče – přesuny hotovosti na účet a z něj.

Bank/default – formulář pro transakce s penězi.

6.3.7 CasinoPresenter

Podává přehled her, které se hrají v casinu.

Casino/default – nabídka her v casinu.

V této práci byla do casina přidána pouze jedna hra, proto tento seznam není zobrazován a je zrovna přístupeno ke hře GOLDtip.

6.3.8 GoldtipPresenter

Tento presenter podává informace o hře GOLDtip. Umožňuje vytvoření hry, připojení se ke hře, zobrazení historie her a pravidel. Obsahuje také zkušební hru proti počítači.

Goldtip/default – vytvořené otevřené hry, ke kterým se lze připojit a zahrát si. V případě že hráč má rozehranou hru, je mu zobrazena tato hra.

Goldtip/played – seznam všech odehraných her v casinu všemi hráči. Zde lze prohlížet strategie ostatních hráčů a snažit se tak zvýšit své šance na výhru.

Goldtip/training – zkušební hra proti počítači.

Goldtip/rules – pravidla hry.

6.3.9 ShopPresenter

Zobrazuje všechny zbraně, zbraně jsou tříděny do kategorií. Obchod umožňuje uživateli zbraň koupit nebo prodat.

Shop/default – seznam všech zbraní, pomocí formuláře lze výběr omezit pouze na určitou kategorii.

Shop/sell – nabídka s vykoupením aktuálně nošené zbraně.

6.3.10 StreetPresenter

Hráč na ulici vidí kolem sebe ostatní hráče, ale pouze na omezenou vzdálenost, tyto postavy tento presenter zobrazuje a umožňuje provést přepadení.

Street/default – seznam hráčů na dohled.

Street/attack/<id> - provede přepadení na hráče s ID = <id>, ověřuje se, zdali je tato postava opravdu v dohledu.

6.3.11 CrimesPresenter

Zobrazuje protokoly přepadení, kde jsou uvedeny další podrobnosti jako čas, útočník, oběť a částka, o kterou byla oběť okradena.

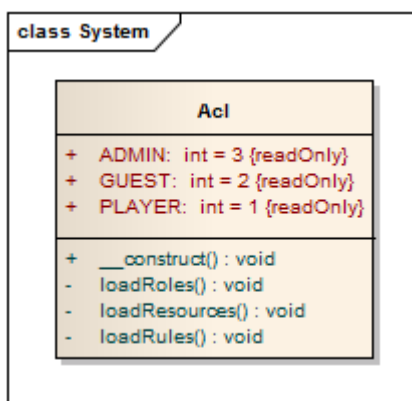
Crimes/default – seznam protokolů o přepadení.

Crimes/details/<id> - podrobnosti o přepadení s ID = <id>.

6.4 Model

Model (viz architektura MVC – Kapitola 2.4) je soubor tříd, které reprezentují objekty ve hře. Třídy mají některé své vlastnosti *private*, jejich hodnoty jsou přístupné pro čtení pomocí tzv. magické metody `__get`. Pro zápis do vlastností objektu je třeba využít některou z veřejných metod třídy.

6.4.1 Třída Acl

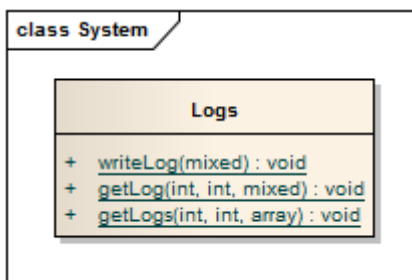


Obrázek 16: Třída Acl

Třída Acl vytváří model přístupových práv uživatelů podle dat uložených v databázi. Vytváří jejich hierarchii podle nadřazenosti jednotlivých rolí a také spravuje jednotlivé zdroje. V aplikaci byly definovány role:

- *Admin* – správce hry
- *Player* – hráč
- *Guest* – nepřihlášený uživatel

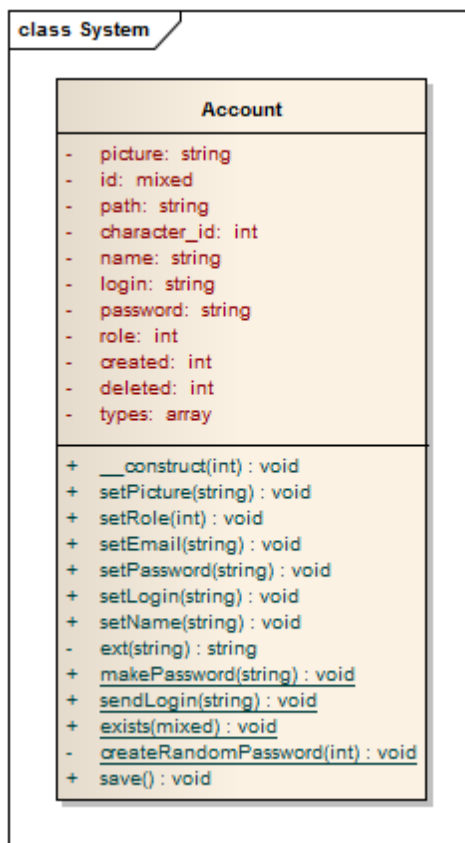
6.4.2 Třída Logs



Obrázek 17: Třída Logs

Třída Logs slouží k ukládání a výběru informací o přihlašovaných uživateli, do databáze se zaznamenává čas přihlášení a IP adresa.

6.4.3 Třída Account

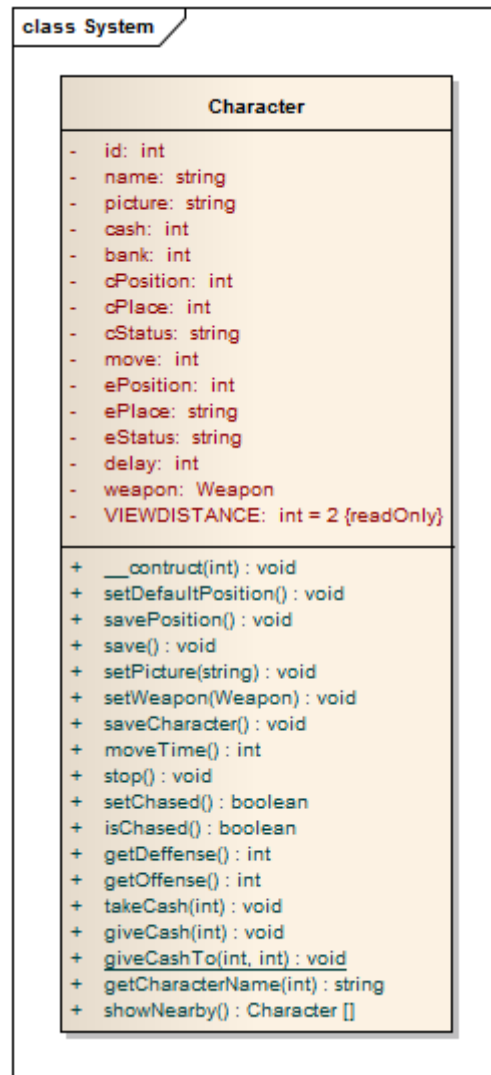


Obrázek 18: Třída Account

Tato třída zapouzdřuje práci s uživatelskými účty, je využita při registraci uživatele nebo při generování, změně nebo zaslání zapomenutého hesla. Pokud se třídě předá v konstruktoru parametr obsahující ID účtu, je údaj načten z databáze. Volání konstruktoru bez parametru vytvoří nový účet, ale do databáze je vložen až po volání metody *save*.

6.4.4 Třída Character

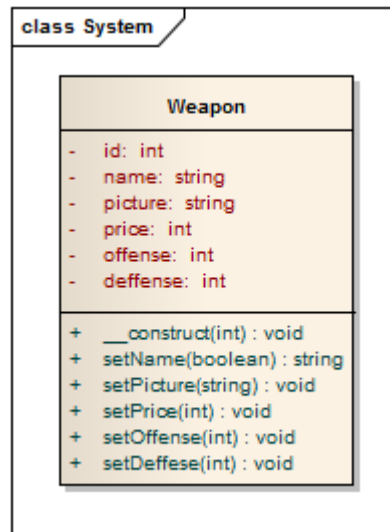
Třída reprezentuje postavu ve hře. Každá postava může mít u sebe jednu zbraň, tu realizuje třída Weapon (Kapitola 6.3.5). Důležitá je též pozice postavy na mapě, postava může stát nebo se pohybovat, být na útěku nebo být třeba v budově. Podrobně je řešení popsáno v Kapitole 6.6.



Obrázek 19: Třída Character

6.4.5 Třída Weapon

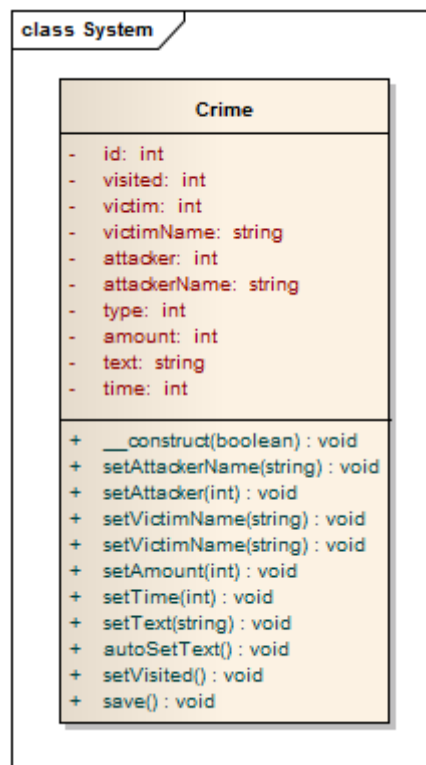
Třída vytváří model zbraně, obsahuje metody pro nastavení vlastnosti zbraně při vytváření objektu z dat získaných z databáze. Útočnou vlastnost zbraně představuje hodnota *offense*, obrannou sílu hodnota *deffense*. Nákupní cenu zbraně stanovuje položka *price*.



Obrázek 20: Třída Weapon

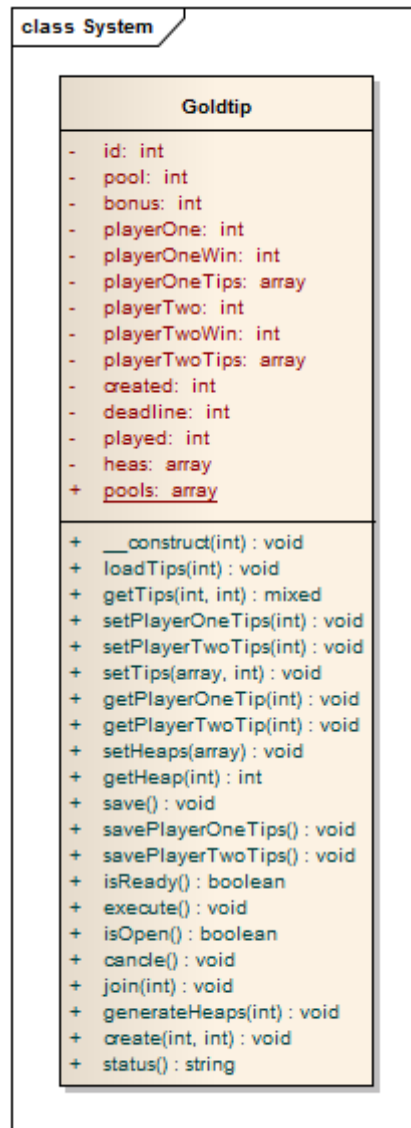
6.4.6 Třída Crime

Konkrétní zápis o přepadení představuje tato třída. Obsahuje časové údaje o události, informace o útočnickovi a poškozeném a taky částku, o kterou oloupený přišel. Položka *visited* indikuje, zda je třeba poškozeného upozornit na to, že byl přepaden. Po shlednutí této informace je zaznamenán čas shlednutí této zprávy.



Obrázek 21: Třída Crime

6.4.7 Třída Goldtip



Obrázek 22: Třída Goldtip

Třída zapouzdřuje celou hru GOLDtip. Poskytuje metody pro vytvoření hry se zvoleným základním vkladem, vygenerování velikostí hromádek a nastavení tipů jednotlivých hráčů a vyhodnocení hry.

__construct(int) : void – konstruktor s nepovinným parametrem ID hry, pokud je parametr zadán při vytváření instance třídy, je objekt nastaven podle hry s odpovídajícím ID v databázi.

private loadTips(int) : void – metoda očekává parametr s ID hráče, tyto tipy vyhledá v databázi a vrací je v podobě pole.

public setPlayerOneTips(array) – nastaví tipy prvního hráče, vlastnost *playerOneTips*.

public setPlayerTwoTips(array) – nastaví tipy druhého hráče, vlastnost *playerTwoTips*.

public setTips(array, int) – nastavení tipů uvedených v prvním parametru hráči s ID ve druhém parametru.

public getPlayerOneTip(int) – vrací velikost tipu prvního hráče, pořadí tipu je předán v parametru funkce.

public getPlayerTwoTip(int) – vrací velikost tipu druhého hráče, pořadí tipu je předán v parametru funkce.

public setHeaps(array) – nastavuje velikost hromádek podle vstupního parametru.

public isReady() – vrací hodnotu typu boolean, která udává, zdali hra obsahuje všechny potřebné data k tomu, aby mohla být vyhodnocena. Musí mít nastaveny hromádky a musí obsahovat tipy obou hráčů.

public execute() – provede vyhodnocení hry.

public cancel() – pokud se již do hry nepřihlásil druhý hráč, je hra zrušena.

public join(int) – připojení hráče do hry, jako parametr je ID hráče.

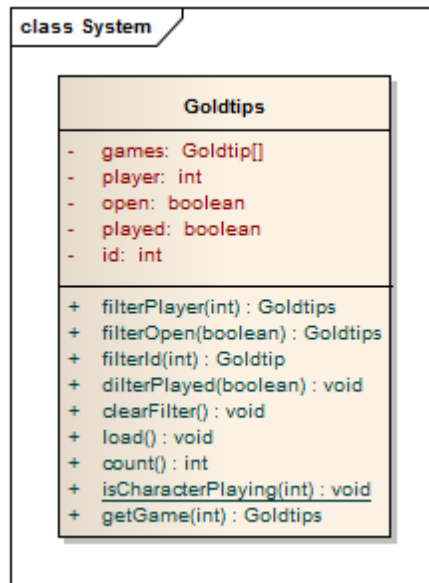
public generateHeaps () – vrací pole s vygenerovanými hodnotami hromádek, při výpočtu vychází ze základního vkladu.

public status() – vrací textový řetězec udávající stav hry, např.: *Čeká se na soupeře*.

public create(int, int) – vytvoří novou hru, první parametr je ID hráče, který hru vytváří, druhý je výše vkladu jednoho hráče. Podle výše vkladu se vypočítá velikost hromádek a velikost bonusu.

public save() – metoda slouží pro uložení aktuálního objektu do databáze. Pokud byl při vytváření instance třídy volán konstruktor s parametrem, je proveden UPDATE, jinak INSERT.

6.4.8 Třída Goldtips

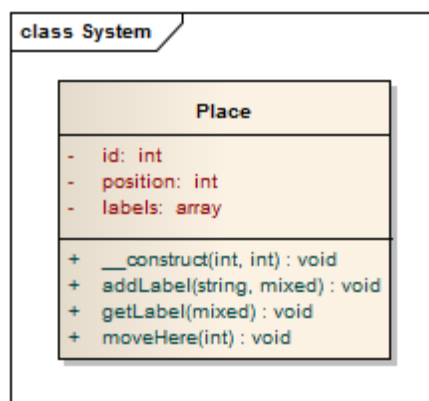


Obrázek 23: Třída Goldtips

Třída obsahuje seznam her, umožňuje nastavovat různé filtry pro získání určité množiny her.

public static isCharacterPlaying(int) – vrací true/false podle toho zda se hráč účastní hry. Tato informace je důležitá z toho důvodu, že pokud hráč hraje, nemůže se zároveň přesunovat po mapě.

6.4.9 Třída Place



Obrázek 24: Třída Place

Tato třída vytváří model místa, kam se může postava vydat. Místo je určeno svým ID, názvem a pozicí. Jelikož český jazyk používá skloňování, bylo umožněno místu dát více

názvů, které si objekt ukládá podle zadaných klíčů. Tyto názvy byly pojmenovány jako *labels*. Podle klíče lze také získat požadovanou *label* místa.

Všechny místa na mapě jsou spravovány pomocí třídy Places (Kapitola 6.3.10). Na Obrázku 25 je zdrojový kód, který přidává do mapy Banku. *Labely* jsou označeny klíči podle jednotlivých pádů, takže klíč *1p* odpovídá názvu v prvním pádu - banka.

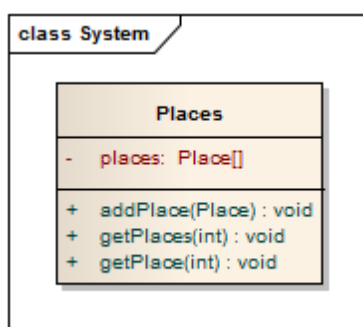
```

23         $this->places = new Places();
24
25         $bank = new Place('Bank', 0);
26         $bank->addLabel('1p', "banka");
27         $bank->addLabel('2p', "banky");
28         $bank->addLabel('3p', "banky");
29         $bank->addLabel('4p', "bance");
30
31         $this->places->addPlace($bank);

```

Obrázek 25: Vytvoření místa Banka

6.4.10 Třída Places



Obrázek 26: Třída Places

Třída obsahuje množinu všech objektů na mapě.

public addPlace(Place) – přidá do mapy místo předané v parametru typu Place (Kapitola 6.4.9).

public getPlaces (int) – vrací všechny dostupné místa v proměnné typu pole, místo předané v parametru je při návratu ignorováno. Tím se získá seznam míst, kam se postava může přesunout.

6.5 Týdenní plat

Každé postavě je na účet připsána částka 500 Buggs týdně, toto zajišťuje jednou za sedm dní vykonávaný skript (Obrázek 27).

```
1 CREATE EVENT mesto_zlodeju_salary
2   ON SCHEDULE
3     EVERY 7 DAY
4     STARTS NOW()
5     COMMENT 'Send salary'
6   DO
7     UPDATE game.characters
8     SET bank = bank + 500;
```

Obrázek 27: Kód zajišťující týdenní plat

6.6 Pozice postavy

Informace o pozici a pohybu každé postavy jsou uloženy v databázi v tabulce *positions*. Na Obrázku 28 je zobrazena struktura této tabulky, PK = Primary Key, NN = Not Null, AI = Auto Increment, UQ = Unique.

Atribut *currentStatus* obsahuje text, popisující aktuální stav postavy, *currentPlace* obsahuje ID místa, kde se právě postava nachází, byly zvoleny textové identifikátory pro lepší orientaci (Bank, Casino, Street, Shop), proto je tento atribut typu VARCHAR(20). Aktuální souřadnice pozice postavy na mapě udává atribut *current*. Pokud se postava nepohybuje, je *move* nastaveno na nulu, při pohybu vlevo je hodnota rovna -1, vpravo 1. Položky *endStatus*, *endPlace* a *end* nesou informace o cílovém místě, kam se postava přesunuje. Dosažení cílové pozice nastane pokud *current* = *end*.

Pokud postava někoho přepadne (Kapitola 4.8.1), musí být na útěku určitý čas. Tento stav značí položka *delay*, která obsahuje počet minut do konce útěku. Pokud je nastavena na hodnotu 0, postava na útěku není a může se normálně pohybovat.

Column Name	Datatype	PK	NN	AI	UQ
id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
character_id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
currentStatus	VARCHAR(40)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
currentPlace	VARCHAR(20)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
current	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
move	TINYINT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
endStatus	VARCHAR(40)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
endPlace	VARCHAR(20)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
end	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
delay	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Obrázek 28: Struktura tabulky *positions*

6.7 Pohyb postavy

Jelikož je potřeba zjišťovat aktuální pozici všech postav (viz Kapitola 4.8.1), je nutné v určitých časových intervalech postavy na mapě posunovat, interval byl zvolen 30 sekund. Tato hodnota je plně dostačující, neboť všechny časové údaje ve hře nejsou uváděny podrobněji nežli v minutách.

Obrázek 18 obsahuje SQL kód, který zajistí vytvoření MySQL Eventu, který se stará o pohyb postav. Na řádce 5 je nastaven časový interval, ve kterém bude prováděn kód mezi značkami BEGIN a END. Příkazy na řádcích 9-12 provádí posun všech postav, které jsou v pohybu a nedošly ještě do cílové pozice. Na řádcích 14 až 22 se kód provádí tehdy, pokud postava dorazila k cíli. Pohyb je zastaven nastavením atributu *move* na 0, aktuální pozice je nahrazena cílovou a té je nastavena hodnota *null*.

Poslední část (řádek 24 až 26) je prováděna u těch postav, které jsou na útěku. U nich je snižován atribut *delay* dokud není nulový.

```
1 delimiter |
2
3 CREATE EVENT mesto_zlodeju
4   ON SCHEDULE
5     EVERY 30 SECOND
6     COMMENT 'Moves users, make delay'
7   DO
8     BEGIN
9       UPDATE game.positions
10      SET current = current + move
11      WHERE move <> 0
12      AND current <> end;
13
14      UPDATE game.positions
15      SET currentStatus = endStatus,
16      currentPlace = endPlace,
17      move = 0,
18      endStatus = NULL,
19      end = NULL,
20      endPlace = NULL
21      WHERE current = end
22      AND delay = 0;
23
24      UPDATE game.positions
25      SET delay = delay - 1
26      WHERE delay > 0;
27 END |
```

Obrázek 29: SQL kód pro posunování postav

Při návrhu řešení posunování postav po mapě bylo uvažováno několik možných způsobů, jak realizovat spouštění skriptu, který by toto zajišťoval. Nabízelo se buď použití Cronu, nebo MySQL Events. Ze dvou způsobů bylo pro své jednoznačné výhody vybráno MySQL Events. Následující přehled rozebírá možné výhody a nevýhody obou řešení.

1. Cron

Cron je samostatný proces, který spouští skripty nebo aplikace v zadaný čas nebo interval. Veškeré úlohy se zadávají do tzv. crontab. Cron by tedy spouštěl skript, který by vytvořil připojení k databázi, spouštěl SQL dotazy a prováděl tak posunování postav. Výhoda tohoto řešení je ta, že se spouští PHP skript, který není omezen pouze na SQL dotazy. Nevýhody tohoto řešení však jsou:

- nutnost vytvořit připojení k databázi

- zpracování PHP skriptu zabere nějaký čas
- je třeba zabezpečit, aby skript nemohl spustit nikdo jiný, než služba Cron

2. MySQL Events

Databáze MySQL umožňuje spouštět SQL dotazy v zadaný čas nebo v daných intervalech. Aby bylo možné využít MySQL Events, je třeba, aby na databázovém serveru byla tato služba spuštěna. Proveďte se to jednoduše nastavením systémové proměnné *event_scheduler* na hodnotu **ON**, v konzole připojené k databázi příkazem **SET GLOBAL event_scheduler = ON;**. Po této operaci je třeba server restartovat.

Výhody tohoto řešení:

- Není potřeba se starat o zabezpečení.

Nevýhodou tohoto řešení může být pouze to, že takto lze provádět pouze SQL dotazy, což je ale v případě této aplikace dostačující.

ZÁVĚR

Cílem této práce bylo navrhnout a vytvořit strategickou tahovou hru v internetovém prohlížeči.

Teoretická část diplomové práce popisuje úvod do teorie her, kde byly definovány základní pojmy tohoto oboru, rozdělení her z různých hledisek a možné způsoby jejich zápisu. Tyto teoretické poznatky byly využity při pozdějším návrhu hry. Dále byly představeny použité technologie a softwarová architektura MVC, kterou podporuje i použitý framework Nette. Vytvořená aplikace byla na této architektuře také postavena, neboť MVC s sebou přináší mnoho nesporných výhod, které jsou v práci zmíněny.

Praktická část popisuje pravidla a herní mechanismy hry. Hra byla navržena pro více hráčů, kteří hrají proti sobě. Pro zprovoznění stačí mít nainstalován webový prohlížeč s podporou JavaScriptu. Hlavní motivací do hry byla uvažována soutěživost mezi hráči o to stát se nejbohatší postavou. Hra byla také rozšířena i o hru proti počítači, ta byla přidána jako zkušební hra GOLDtip, jež je podrobně rozebrána v Kapitole 5, kde je uveden i příklad konkrétní partie s podrobně rozepsanými strategiemi obou hráčů.

Výsledkem praktické části je funkční aplikace s názvem Město zlodějů. Ke zdrojovému kódu napsanému v jazyce PHP byla vytvořena podrobná dokumentace, kterou jistě ocení ten, kdo by chtěl aplikaci dále rozšířit nebo upravit. Návod k instalaci Města zlodějů je uveden v Příloze II. Na přiloženém kompaktním disku jsou k dispozici veškeré zdrojové soubory a také kód SQL pro vytvoření základních tabulek databáze.

Aplikace byla testována na lokálním serveru a chtěl jsem ji otestovat i na některém webovém hostingu zdarma. Zvolené řešení pohybu postav ale vyžaduje podporu MySQL Events, což je služba, která nebývá u hostingů zdarma k dispozici. Nabízí se tedy druhé řešení – cron, ale mé další zjištění bylo, že ani služba cron se u hostingů zdarma nevyskytuje často a už stěží dovoluje spouštět skripty v třicetisekundových intervalech. Pro zprovoznění hry veřejně bude tedy potřeba využít placeného hostingu.

Vytvořit originální a zábavnou hru tak, aby u ní hráči vydrželi dlouhou dobu, není jednoduché. Úspěch závisí na spoustě věcí – na přitažlivé grafice, tématu hry, na originálním nápadu a zvolené technologii. Psát hry ve webovém prohlížeči je dle mého názoru výhodné zejména proto, že lze velmi rychle dosáhnout velkého rozšíření, a rozvoj internetu a webových technologií tomu jen přispívá.

ZÁVĚR V ANGLIČTINĚ

The aim of this thesis was to design and develop a strategic turn-based game in a web browser.

The theoretical part of the thesis describes an introduction to game theory, which defined the basic concepts of this field, the distribution of games from different perspectives and possible ways of their notes. These theoretical findings were later used to design the game. Used technologies and software architecture called MVC, that the framework Nette supports, were also introduced. Created application was built also on this architecture because MVC has many advantages, which are mentioned in the thesis.

The practical part describes the rules and designed mechanisms of game play. The game was designed for more players who play against themselves. To launch the game it is sufficient to have installed a web browser with JavaScript support. The main motivation for the players was considered competition between players to become the richest character. The game has also been extended to allow playing against the computer, it was added as the training game of GOLDtip, which is described in detail in Chapter 5. The normal form of this game was used to describe the strategies of each player of game GOLDtip.

The result of the practical part is a functional application called City of Thieves For the source code written in PHP language was created detailed documentation that will be certainly appreciated those who would like to further extend or modify the application. Installation instructions Cities of Thieves are given in Appendix II.

The application has been tested on a local server and I wanted to test it on a free web hosting. But the chosen solution of moving characters requires the support of MySQL Events, a service that is not usually supported on free hostings. However, there is the second solution - cron. My next finding was that even free hostings do not support this service so often. To launch the game will need to use a paid hosting.

Create an original and entertaining game so make players stayed on it for a long time, is nothing simple. Success depends on many things - the attractive graphics, the game theme, the original idea and selected technology. Making games in a web browser, in my opinion, are especially useful because it can very quickly reach a large expansion and development of Internet and Web technologies it just helps.

SEZNAM POUŽITÉ LITERATURY

- [1] Gamedesign.cz [online]. Vývoj počítačových her obecně. Dostupné z WWW: [http://gamedesign.cz].
- [2] Rutledge, M. PHP game programming. Thoson course technology. ISBN: 1-59200-153-X.
- [3] Morrison, M. Naučte se programovat počítačové hry za 24 hodin. ComputerPress, 2004, ISBN 80-251-0371-4.
- [4] Top 1 to 50 Games and Entertainment - PHP scripts at Programsdb.com [online]. Game scripts. Dostupné z WWW: [http://www.programsdb.com/scriptscategory/10/44/1/0.html].
- [5] Straffin, P.D.: Game Theory and Strategy, The Mathematical Association of America, 2003.
- [6] Dresher, M.: The Mathematics of Games of Strategy, Theory and Applications, Dover Publications, 1981.
- [7] Hykšová, M. Teorie her a optimální rozhodování, výukové materiály [online], Dostupné z WWW: [http://euler.fd.cvut.cz/predmety/teorie_her].
- [8] *MySQL : Developer Zone* [online]. 2010 [cit. 2011-05-21]. MySQL 5.6 Reference Manual. Dostupné z WWW: <http://dev.mysql.com/doc/refman/5.6/en/index.html>.
- [9] Grudl, David. *Zdroják : tvorba webových stránek a aplikací* [online]. 2009 [cit. 2011-05-21]. Seriál Začínáme s Nette Framework. Dostupné z WWW: <http://zdrojak.root.cz/serialy/zaciname-s-nette-framework/>.
- [10] Myerson, Roger B. *Game theory: analysis of conflict*. Cambridge (Massachusetts) : Harvard University Press, 1997. 568 s. Dostupné z WWW: <http://books.google.com/books?id=E8WQFRCsNr0C>. ISBN 0-674-34116-3.
- [11] Tichý, Jan. *Programová podpora tvorby webových aplikací*. Praha, 2004. 66 s. Diplomová práce. Vysoká škola ekonomická v Praze, Fakulta informatiky a statistiky.

- [12] Vrána, Jakub. *NotORM : PHP library for simple working with data in the database* [online]. 2010 [cit. 2011-05-23]. Dostupné z WWW: <<http://www.notorm.com>>.
- [13] Pytlíčková, Lenka. *Elementy teorie her*. Olomouc, 2008. 52 s. Bakalářská práce. Univerzita Palackého v Olomouci, Přírodovědecká fakulta, Katedra matematické analýzy a aplikací matematiky. Dostupné z WWW: <<http://mant.upol.cz/soubory/OdevzdanePrace/B08/b08-05-lp.pdf>>.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

PHP	PHP Hypertext Preprocesor
ASP.NET	Active Server Pages .NET
HTML	HyperText Markup Language
SQL	Structured Query Language
GNU	GNU's Not Unix
GPL	General Public Licence
NotORM	Not Object-Relational Mapping
ISAM	Indexed Sequential Access Method
AJAX	Asynchronous Javascript And XML
AJAJ	Asynchronous Javascript And Jason
MVC	Model View Controller
BSD	Berkeley Software Distribution
PDO	PHP Data Objects
DOM	Document Object Model
ER model	Entity Relationship Model
IP	Internet Protocol
URL	Uniform Resource Locator
JSON	JavaScript Object Notation
UML	Unified Modeling Language

SEZNAM OBRÁZKŮ

Obrázek 1: Příklad hry Kámen-nůžky-papír v explicitním tvaru.....	13
Obrázek 2: Příklad hry Kámen-nůžky-papír v normálním tvaru	14
Obrázek 3: Příklad optimální strategie v maticové hře [7]	17
Obrázek 4: Architektura Model-View- Presenter [9]	20
Obrázek 5: Architektura MySQL [8]	21
Obrázek 6: Schéma fungování NotORM [9]	22
Obrázek 7: jQuery – Datepicker	23
Obrázek 8: jQuery – Dialog	24
Obrázek 9: Plán města	29
Obrázek 10: Partie hry GOLDtip v normálním tvaru	38
Obrázek 11: GOLDtip – ukázka partie	40
Obrázek 12: Případy užití	41
Obrázek 13: Funkce pro kontrolu existence účtu	42
Obrázek 14: Rozdělení na presentery	43
Obrázek 15: Metoda pro zpracování formuláře přihlášení uživatele.....	44
Obrázek 16: Třída Acl	47
Obrázek 17: Třída Logs	47
Obrázek 18: Třída Account.....	48
Obrázek 19: Třída Character.....	49
Obrázek 20: Třída Weapon.....	50
Obrázek 21: Třída Crime	50
Obrázek 22: Třída Goldtip	51
Obrázek 23: Třída Goldtips	53
Obrázek 24: Třída Place.....	53
Obrázek 25: Vytvoření místa Banka	54
Obrázek 26: Třída Places	54
Obrázek 27: Kód zajišťující týdenní plat.....	55
Obrázek 28: Struktura tabulky <i>positions</i>	56
Obrázek 29: SQL kód pro posunování postav	57

SEZNAM TABULEK

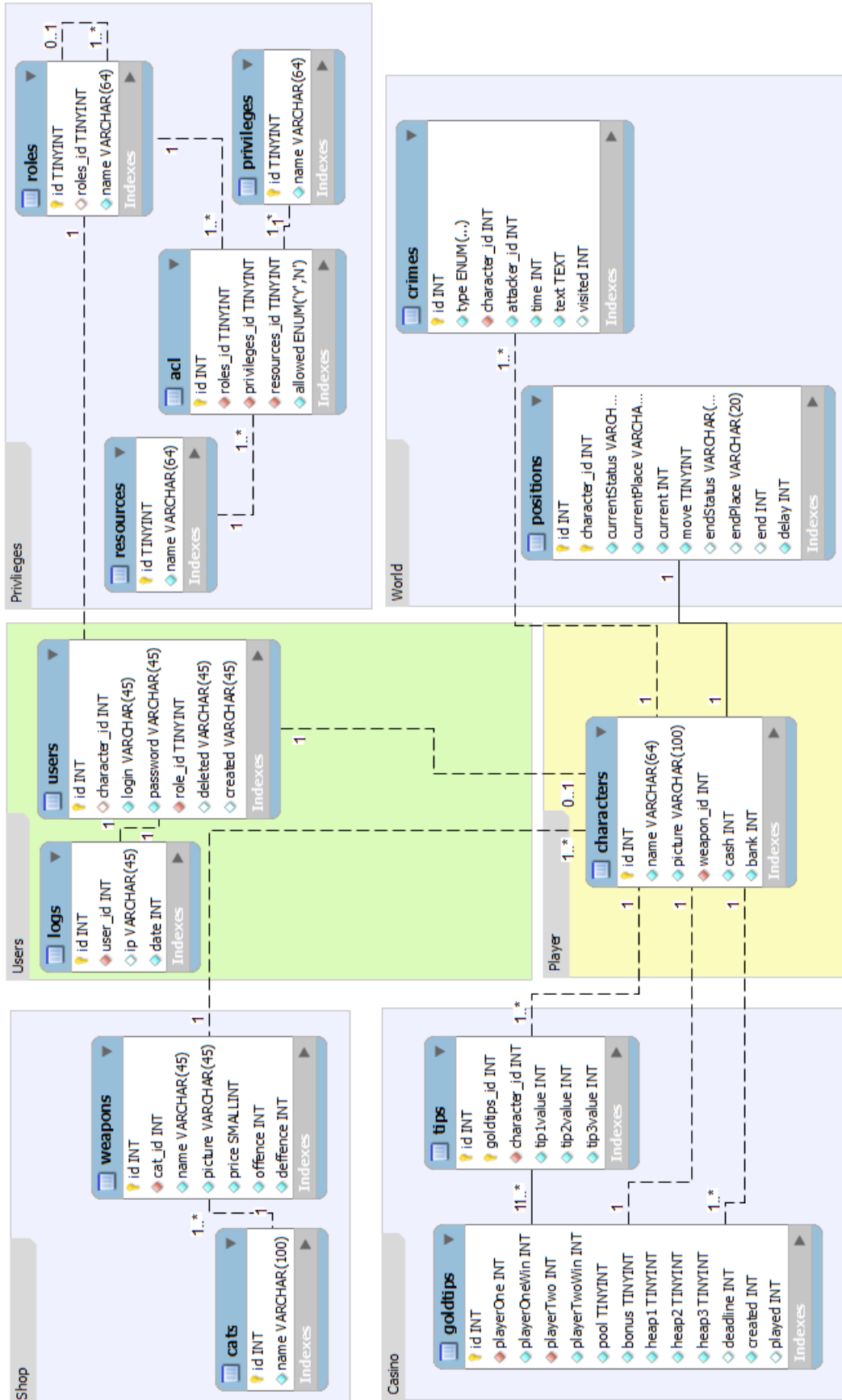
Tabulka 1: Vzdálenosti mezi budovami	29
Tabulka 2: Přehled druhů holí	31
Tabulka 3: Přehled druhů praků.....	31
Tabulka 4: Přehled druhů dýk.....	32
Tabulka 5: Přehled druhů mečů	32
Tabulka 6: Přehled seker.....	33
Tabulka 7: Přehled druhů řemdihů	33
Tabulka 8: Přehled druhů palcátů	34
Tabulka 9: Přehled druhů štítů.....	34
Tabulka 10: Přehled druhů zbroje.....	35
Tabulka 11: Příklad přepadení	36
Tabulka 12: Příklad hry Goldtip se společným bankem 2000 Buggů.....	38

SEZNAM PŘÍLOH

PI – Databázový model

PII – Instalace aplikace

PŘÍLOHA P I: DATABÁZOVÝ MODEL



PŘÍLOHA P II: INSTALACE APLIKACE

Součásti instalace

Na přiloženém disku se nachází adresář instalace se dvěma podadresáři aplikace a databáze, obsah obou je popsán níže.

V adresáři aplikace se nachází podadresáře:

app

Obsahuje třídy modelu, presentery, šablony *.latte*, komponenty využívané v aplikaci a konfigurační soubor aplikace *config.ini*. V něm se nastavuje připojení k databázi a určuje se cesta k adresářům pro tzv. RobotLoader, jenž se stará o automatické načítání tříd.

libs

Zde jsou uloženy knihovny frameworku Nette a databázového layoutu NotORM.

logs

Tento adresář slouží pro vypisování chyb aplikace, což je velmi výhodné při pozdějším ladění programu.

temp

Vygenerované views ukládá framework právě zde.

www

Tento adresář obsahuje soubor *index.php*, který spouští celou aplikaci. Adresář obsahuje také JavaScriptové soubory, styly a obrázky.

V adresáři databáze se nachází soubory s SQL příkazy pro vytvoření základního obsahu databáze:

databaze.sql – vytvoření tabulek

move.sql – vytvoří MySQL Event realizující pohyb postav

salary.sql – vytvoří MySQL Event realizující týdenní výplatu

Postup instalace na serveru

Ke zprovoznění aplikace je třeba mít nainstalovaný server Apache 2, PHP minimálně ve verzi 5.2 a MySQL server od verze 5.1. Balíček XAMPP, který obsahuje všechny výše zmíněné funkce je zdarma dostupný na adrese [HTTP://WWW.APACHEFRIENDS.ORG/EN/XAMPP.HTML](http://www.apachefriends.org/en/xampp.html).

Aplikace po nainstalování obsahuje uživatele s rolí admin, přihlašovací jméno uživatele je *admin*, heslo *admin*. Obsahuje také uživatele s rolí player, přihlašovací jméno *player*, heslo *player*.

Postup instalace v operačním systému Windows:

1. Na přiloženém disku otevřete adresář instalace.
2. Vytvořte databázi s názvem *game* (tento název je nutné dodržet) a spusťte nad databází dotazy uložené v souborech: *databaze.sql*, *move.sql* a *salary.sql*. Oba soubory jsou umístěny v adresáři databaze.
3. V konfiguračním souboru server MySQL *my.ini* (nachází se v *c:\xampp\mysql\bin*) vložte následující příkaz, který zapne službu Events: `event-scheduler = ON`. Restartujte MySQL server.
4. Zkopírujte obsah adresáře aplikace na server tak, aby složka *www* byla veřejně přístupná, tzn., aby se nacházela v adresáři nastaveném jako *document_root* v konfiguraci serveru. Ostatní složky (*app*, *logs*, *libs* a *temp*) naopak nesmí být veřejně přístupné, k tomu je využito souborů *.htaccess* v každém s adresářů. Je třeba mít na serveru tuto funkci zapnutou.
5. V adresáři *app* se nachází konfigurační soubor *config.ini*. V něm je třeba nastavit adresu MySQL serveru, přihlašovací jméno a heslo.
6. Aplikaci spustíte otevřením souboru *index.php* ve webovém prohlížeči.

Tento postup byl otestován na Windows XP Professional SP3 a s nainstalovaným balíčkem XAMPP 1.7.4.