

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

KLIENT PRO SPRÁVU HERNÍCH SERVERŮ

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

TOMÁŠ CIGÁN

BRNO 2010



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

KLIENT PRO SPRÁVU HERNÍCH SERVERŮ

GAME SERVER ADMINISTRATION CLIENT

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

TOMÁŠ CIGÁN

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. TOMÁŠ KAŠPÁREK

BRNO 2010

Abstrakt

Tato bakalářská práce se zaměřuje na popis herních serverů pro nejhranější herní tituly. Podrobně jsou rozebrány rozdíly mezi jednotlivými herními servery a možnostmi jejich ovládání. Dále je zde rozebrána struktura a způsob instalace pluginů na tyto servery. Z předchozího rozboru byl navržen a implementován systém pro zautomatizování a jednotnou správu těchto rozšíření a ovládání ve formě webové aplikace. Je zde rovněž předveden způsob rozšíření podpory o nové herní servery.

Abstract

This bachelor thesis focuses on the description of game servers of the most common computer games. The details and differences between particular game servers as well as administration control are included. Also structure of plugins and plugin installation procedures for such servers is described in details. The main target of the thesis according to the previous research is to analyze and develop uniform system for controlling the game server and its plugins thru web application. At the end of the thesis there is detailed description of creating a new game server implementation support.

Klíčová slova

herní servery, ovládání herních serverů, správa, administrace, instalace pluginů, server, hry, modifikace, pluginy

Keywords

game servers, game server control, administration, plugins installation, server, games, mods, modifications, plugins

Citace

Tomáš Cigán: Klient pro správu herních serverů, bakalářská práce, Brno, FIT VUT v Brně, 2010

Klient pro správu herních serverů

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Tomáše Kašpárka

.....
Tomáš Cigán
18. května 2010

Poděkování

Touto cestou bych chtěl poděkovat vedoucímu své bakalářské práce, Ing. Tomáši Kašpárkovi, za umožnění realizace a odborné vedení této práce.

© Tomáš Cigán, 2010.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1 Úvod	4
2 Úvod do problematiky	5
2.1 Základní možnosti ovládání a správy herních serverů	5
2.1.1 Dotazování - query	5
2.1.2 Vzdálená konzole - rcon	6
2.1.3 Úprava souborů - FTP	6
2.1.4 Vzdálené volání procedur - XML-RPC	6
2.1.5 Speciální aplikace - GSDaemon	6
3 Rozbor herních serverů	7
3.1 GoldSource Engine	7
3.1.1 Query	7
3.1.2 Rcon	8
3.1.3 Pluginy	9
3.2 Source engine	10
3.2.1 Query	10
3.2.2 Rcon	10
3.2.3 Pluginy	11
3.3 Id tech engine	12
3.3.1 Query	12
3.3.2 Rcon	13
3.3.3 Pluginy	13
3.4 Série her Battlefield	13
3.4.1 Query	14
3.4.2 Rcon	15
3.4.3 Pluginy	15
3.5 Unreal engine	15
3.5.1 Query	16
3.5.2 Rcon	16
3.5.3 Pluginy	16
3.6 Série Trackmania	16
3.6.1 Query a ovládání	17
3.6.2 Pluginy	17

4 Implementace	18
4.1 Databázový návrh a implementace	18
4.2 Hlavní plugin	20
4.2.1 Dotazovací protokoly herních serverů	20
4.2.2 Vzdálené ovládání pomocí rcon protokolu	21
4.2.3 Detekce pluginů na serveru	22
4.2.4 Instalace pluginů	23
4.3 Uživatelské rozhraní	24
4.3.1 Seznam serverů	25
4.3.2 Přidávání pluginů do systému	25
4.3.3 Ovládání serveru	25
4.3.4 Instalace a aktualizace pluginů	26
4.4 Bezpečnost	26
4.4.1 Autentizace	26
4.4.2 Autorizace	26
5 Demonstrace rozšíření podpory her	27
5.1 Query modul	27
5.2 Rcon modul	27
5.3 Modul detekce pluginů	28
6 Závěr	29
6.1 Další vývoj	29
6.2 Zhodnocení odvedené práce	29

Seznam obrázků

3.1	Komunikace rcon protokolu	9
3.2	Struktura serveru s nainstalovanými pluginy	10
3.3	Source rcon protokol	11
3.4	Typický případ mnoha zanořených pluginů	11
3.5	Příklad komunikace	15
3.6	Struktura pluginů hry Trackmania	17
4.1	Databázové schéma tříd	19
4.2	Modul dotazovacích protokolů	21
4.3	Modul vzdáleného ovládání pomocí rcon protokolu	22
4.4	Modul detekce pluginů na serveru	23
4.5	Grafické rozhraní	24
5.1	Adresářová struktura hlavního pluginu	27

Kapitola 1

Úvod

V dnešní době stále přibývajících hráčů počítačových her vyvstává potřeba stále více herních serverů. Přibývá také více poskytovatelů těchto serverů, takže není problém si vlastní herní server objednat podobně jako například webhosting. Hry začínají hrát stále mladší a mladší lidé, kteří však nemají zkušenosti s provozem a správou vlastního serveru. Tato práce se tedy bude zabývat návrhem aplikace, která zjednoduší proces správy a ovládání a bude obsahovat návrh systému pro snadnou instalaci rozšíření herního serveru. Budeme se zde tedy zabývat pouze hrami, které podporují hru více hráčů po síti a k dispozici mají volně stažitelný dedikovaný server. Jelikož her, které tyto parametry splňují je nepřehledné množství a není možné je v této práci všechny obsáhnout, budeme rozebírat pouze ty nejpoužívanější.

Kapitola 2

Úvod do problematiky

2.1 Základní možnosti ovládání a správy herních serverů

Možností, jak ovládat server pouze pomocí základních nástrojů v něm zahrnutých, je hned několik. Každý server v sobě implementuje možnost získat z něj základní údaje jako je například název serveru a počet aktuálně hrajících hráčů. K tomuto slouží protokol jednoduše nazvaný *query*. Slouží ovšem pouze k získání informací ze serveru a není pomocí něj možné měnit žádné parametry serveru. Tento protokol však není nijak standardizovaný a každý typ hry jej má implementový jinak. Nejčastěji jsou servery vzdáleně ovládány pomocí *rcon* protokolu, který je podrobně popsán v kapitole 2.1.2 na straně 6. Ten bývá stejně jako protokol předcházející implementován v různých formách pro většinu herních serverů.

Další a nejčastější možností správy serveru je protokol *FTP*, pomocí kterého můžeme na serveru měnit konfigurační soubory, přidávat různé pluginy, případně nové mapy. Ke každému serveru až na výjimky je umožněn přístup přes protokol *FTP*. Poslední možností ovládání je použití speciálního démona, pod kterým je server spuštěn.

2.1.1 Dotazování - query

Pomocí tohoto protokolu je možno ze serveru získat různorodé informace o aktuálním stavu, ve kterém se právě nachází. Jak již bylo zmíněno výše, tento protokol není nijak standardizován a každá hra, případně celá série her od stejných autorů jej implementuje jinak. To vede k tomu, že informace, které jsme schopni tímto protokolem o serveru získat, se liší dle typu hry. Existuje však několik málo parametrů, jenž jsou společné prakticky pro všechny typy herních titulů. Mezi ty nejdůležitější patří

- název serveru,
- aktuální počet hráčů,
- maximální počet hráčů,
- aktuální mapa,
- typ hry,
- volně přístupný/zaheslovaný.

Parametrů, které můžeme získat je samozřejmě mnohem více. Často je možné tímto způsobem ze serveru dostat například seznam všech hráčů právě hrajících, případně seznam všech pravidel

neboli proměnných, které jsou nastaveny. Podrobně se tímto tématem budeme zabývat v kapitole 3 na straně 7 pro každý typ serveru zvlášť.

2.1.2 Vzdálená konzole - rcon

Tento protokol slouží především k jednoduché správě serveru pomocí *rcon* příkazů. Zkratka *rcon* pochází z anglického výrazu „Remote CONTROL“, což se dá přeložit jako vzdálené ovládání. Podobně jako v předchozím případě, ani zde neexistuje žádná norma pro to, jak má tento protokol standardně vypadat. Princip je však téměř u všech herních titulů velice podobný.

Základem je znalost takzvaného *rcon* hesla, bez kterého není možné s tímto protokolem pracovat. Toto heslo se nastavuje u každého serveru v jeho hlavním konfiguračním souboru. Abychom mohli serveru posílat příkazy přes tento protokol, je třeba se nejdříve tímto heslem na serveru autorizovat. Poté můžeme serveru zasílat libovolné příkazy přímo ze hry. Nejjednodušší způsob je tedy otevřít si ve hře příkazový řádek (často pomocí klávesy ~), autorizovat se heslem a poté zadávat jednotlivé příkazy. Tato metoda se často využívá například k násilnému odpojení nechtěných hráčů ze hry administrátorem. K zadávání těchto příkazů však není nutné být připojen ve hře. Můžeme například využít programů třetích stran, které tento protokol umí používat. Mezi nejznámější a nejpoužívanější patří aplikace HLSW¹, která slouží jako adresář oblíbených serverů. Podrobně se tímto protokolem budeme zabývat jednotlivě pro každý herní server v kapitole 3 na straně 7.

2.1.3 Úprava souborů - FTP

Slouží pro přenos souborů z a na server. U drtivé většiny provozovatelů herních serverů je přístup k souborům přes protokol FTP samozřejmostí. FTP je nutné pro nahrávání pluginů a ke změně konfiguračních souborů, jelikož pomocí *rcon* protokolu se změněné parametry nikdy do konfiguračních souborů neukládají. Pokud bychom je tedy změnili pouze příkazem přes *rcon* protokol, po restartu serveru by tyto provedené změny pozbyly platnosti.

2.1.4 Vzdálené volání procedur - XML-RPC

Pomocí *RPC* je možno provádět vzdálené volání procedur. *XML*, tedy *eXtensible markup language* tato volání pouze zapouzdřuje. Protokol *XML-RPC* je přenášen pomocí protokolu *http* a je tedy možné zajistit komunikaci mezi různými operačními systémy a jejich aplikacemi. Tento protokol však podporuje naprosté minimum herních titulů. Problematika byla čerpána z [8].

2.1.5 Speciální aplikace - GSDaemon

Tento démon je schopen zastat funkci FTP a *rcon* protokolu dohromady a přidat další zajímavé možnosti ovládání serveru jako je například živý výpis z konzole serveru, možnost psaní příkazů přímo do konzole serveru a další. Tato aplikace, nebo-li démon, je vyvíjen v rámci bakalářské práce Janem Faikou [1]. V této aplikaci bude použita pouze knihovna *GSClient* dodaná autorem aplikace *GSDaemon*, s pomocí které bude aplikace s démonem komunikovat.

¹ domovská stránka programu: <http://www.hls.w.org/>

Kapitola 3

Rozbor herních serverů

Herní servery pro účel této aplikace rozdělíme do skupin podle jejich možností konfigurace a ovládání. Herní servery sdílející stejný engine¹ mají téměř vždy stejný způsob jejich konfigurace a stejný způsob instalace pluginů. Je tedy vhodné je rozdělit právě podle engine, na kterém hra běží. Další možností rozdělení je podle celé série her, pokud má daná hra více dílů. Podrobné rozdělení her podle herních engine můžeme například najít v [3]. Je zde taktéž dobře ukázána hierarchie herních engine a jejich dědění.

3.1 GoldSource Engine

Mezi hráči často známější pod názvem Half-life 1, patří mezi nejoblíbenější a nejrozšířenější hru na světě. Dedikovaný server pro tuto hru je dostupný jak pro operační systém Windows, tak pro Linux a po drobných úpravách i pro systém FreeBSD. Data všech dedikovaných serverů níže uvedených je možno stáhnout pomocí aplikace *steam*. Herní server samotný rovněž tuto aplikaci používá k automatickým aktualizacím, jsou-li uživatelem nastaveny.

Jedná se o silně modifikovaný *QuakeWorld* engine, který se používal ve hrách Quake a jejímž autorem je studio Id-tech. Autorem tohoto engine a taktéž všech níže vyjmenovaných her je *Valve*.

Mezi nejčastější hry, které tento engine využívají patří například:

- Half-Life Deathmatch
- Counter-Strike
- Day of Defeat
- Team Fortress Classic

3.1.1 Query

Query protokol pro tuto hru je implementován na síťovém protokolu UDP/IP. Port, na který je nutno požadavek zaslat, je stejný jako herní port. V prvním kroku je třeba serveru zaslat následující řetězec:

```
\xFF\xFF\xFF\xFFSource Engine Query\x00
```

Jako odpověď poté dostaneme paket s hlavičkou obsahující:

¹Engine je softwarový prostředek pro tvorbu počítačových her

\xFF\xFF\xFF\xFF

a tělem paketu obsahujícím posloupnost znaků. Tyto znaky je nutno dekodovat na různé datové typy, kterých je celkem 7. Těmito datovými typy jsou:

- **char** - znak
- **byte** - 8 bitový integer
- **short** - 16 bitový integer
- **long** - 32 bitový integer
- **float** - 32 bitové reálné číslo
- **long long** - 64 bitový integer
- **string** - řetězec proměnné délky ukončený znakovou nulou 0x00

Tělo paketu obsahuje informace dle níže uvedené tabulky 3.1.

Hodnota	Datový typ	Význam
Typ	char	0x49
Verze	string	Verze síťového protokolu
Název Serveru	string	Název herního serveru
Mapa	string	Aktuálně spuštěná mapa
Složka hry	string	Cstrike pro hru counter-strike
Popis hry	string	Název hry
AppID	byte	ID aplikace v programu steam
Počet hráčů	byte	Počet aktuálně připojených klientů na serveru
Počet slotů	byte	Maximální možný počet klientů na serveru
Dedikovaný	char	Příznak dedikovaného serveru
OS	char	Operační systém
Zaheslovaný	byte	Vyžaduje heslo?
url_info	string	Adresa s informacemi o tomto módu
url_down	string	Adresa ke stažení tohoto módu
Extra data	string	Pokud je nastaveno, pak jsou dostupné i další parametry
Nevyužito	string	Nevyužito
verze modu	long	Verze
velikost modu	long	Velikost pro stažení
sv_only	byte	Mód je pouze pro server
cidll	byte	Mění klientské dll

Tabulka 3.1: Tělo query paketu

Tuto problematiku jsem studoval z [5].

3.1.2 Rcon

Tento protokol funguje opět na síťovém protokolu UPD/IP s portem, který je stejný jako port pro připojení do hry. V prvním kroku je nutno zaslat řetězec, který obsahuje následující informace:


```
\xFF\xFF\xFF\xFFchallenge rcon\n\x00
```

Tímto požadavkem si zažádáme o unikátní identifikační číslo, kterým prokazujeme svou identitu serveru. Odpověď na předchozí dotaz s vyžádaným číslem tedy přijde ve tvaru:

```
\xFF\xFF\xFF\xFFlchallenge rcon id\n\x00
```

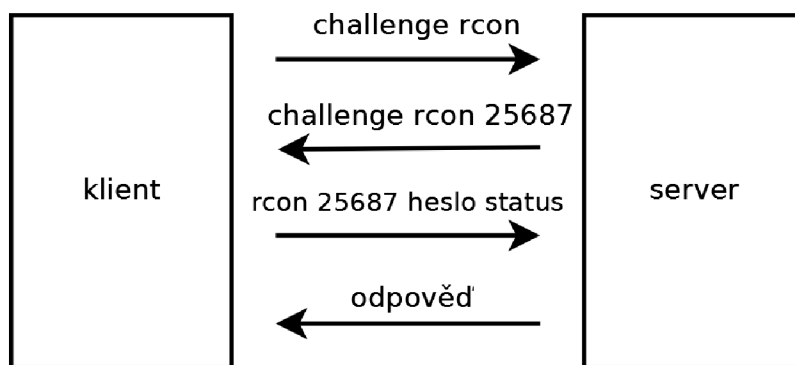
Pokud jsme úspěšně získali identifikační číslo, můžeme začít posílat příkazy. K tomu je nutno znát tzv. *rcon* heslo viz. kapitola 2.1.2 na straně 6. Tyto příkazy posíláme ve tvaru:

```
\xFF\xFF\xFF\xFFrcon id heslo příkaz\n\x00
```

a odpovědi na tyto příkazy poté přijdou ve tvaru:

```
\xFF\xFF\xFF\xFFlodpověď\n\x00
```

Komunikace tedy může probíhat způsobem popsáním na obrázku 3.1.



Obrázek 3.1: Komunikace rcon protokolu

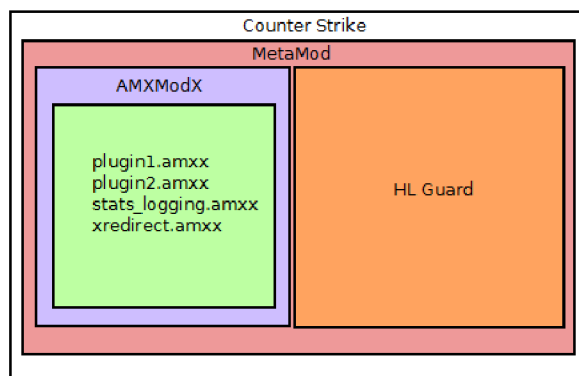
Informace o tomto protokolu jsem čerpal z [6].

3.1.3 Pluginy

Tato hra je výjimečná počtem modifikací², které pro ni existují. Pluginů napsaných pro tuto hru existuje několik set až tisíc. Pluginy se u této hry instalují pouze na server, který poté klientům pouze rozesílá zpracovaná data. Pokud daný plugin pro svou funkci potřebuje například přídavné zvuky, případně další textury či modely, klient si je automaticky ze serveru po připojení stáhne.

Základním pluginem nebo spíše knihovnou pro spouštění dalších pluginů pro tuto hru je *Meta-mod*. Tento plugin je ojedinělý, jelikož jej nelze nahradit žádným jiným. Existuje pouze jeho optimalizovaná, avšak plně kompatibilní verze s názvem *Metamod-p*. Bez tohoto pluginu není možné na server nahrát jakékoliv další modifikace, jelikož jsou na něm všechny další pluginy závislé a pro svou funkčnost jej bezpodmínečně vyžadují. Běžná struktura serveru s nainstalovanými pluginy je zobrazena na obrázku 3.2.

²stránka věnující se těmto pluginům <http://www.amxmodx.org/>



Obrázek 3.2: Struktura serveru s nainstalovanými pluginy

3.2 Source engine

Tento engine je novým a vylepšeným potomkem výše zmíněného *GoldSource* engine. Vznikl v roce 2004 a byl vytvořen stejným vývojovým studiem jako jeho předchůdce. Hry na něm založené se staly velmi populární, avšak doposud co do počtu hráčů nedokázaly překonat svou starší verzi.

Hra i samotný dedikovaný server doznal mnoha změn, které mají především zjednodušit tvorbu modifikací pro tuto hru. Engine je především známý svým, na svou dobu, unikátním fyzikálním modelem.

Dedikovaný server je opět dostupný pro všechny běžně používané operační systémy a je vytvářen přímo samotnými autory hry. Data dedikovaného serveru je opět možno stáhnout přes program zvaný *Steam*, pomocí kterého je taktéž možné server udržovat v aktualizovaném stavu. Mezi hry využívající tento engine patří:

- Half-Life 2: Deathmatch
- Counter Strike: Source
- Day of Defeat: Source
- Team Fortress 2
- Left4Dead
- Left4Dead 2

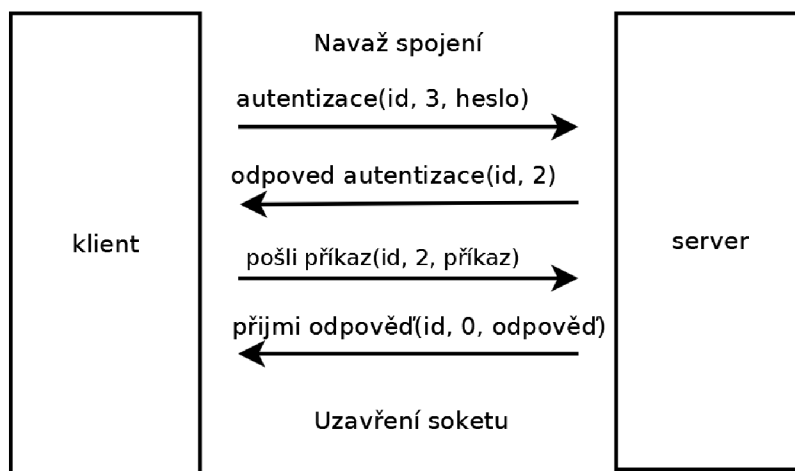
3.2.1 Query

Query protokol tohoto typu her je naprosto totožný s protokolem předcházejícího engine v kapitole 3.1.1 na straně 7.

3.2.2 Rcon

Protokol rcon je oproti předcházejícímu *GoldSource* engine implementován nad síťovým protokolem TCP/IP. Port, na který se budeme připojovat, je opět stejný jako herní port. V prvním kroku je tedy nutné navázat TCP spojení se serverem. Poté je třeba se vůči serveru autentizovat pomocí hesla. Zde je patrný rozdíl oproti předchozí verzi, kdy bylo heslo nutno zasílat s každým příkazem. Zaslání

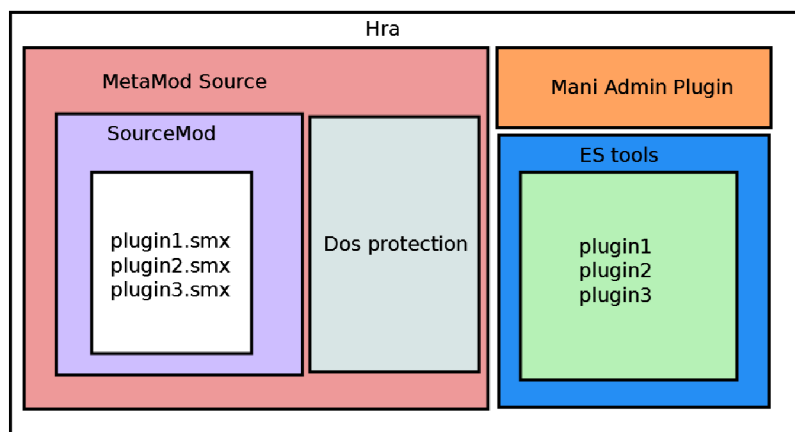
hesla tedy proběhne pouze na začátku a poté již můžeme serveru zasílat příkazy a čekat na odpovědi. Všechny řetězce, které posíláme na server, jsou ukončeny dvěma znakovými nulami. Stejně je to i pro řetězce ze serveru přijímané jako odpovědi. Komunikace je popsána na obrázku 3.3.



Obrázek 3.3: Source rcon protokol

3.2.3 Pluginy

Tento engine se ve struktuře pluginů oproti předchozímu liší především v tom, že již nepotřebuje jeden speciální plugin, pod kterým běží všechny ostatní. To s sebou však přináší nepříjemnosti v neunifikovanosti těchto pluginů. Zatímco v předchozí verzi byly všechny pluginy tvořeny v jednom *API*³, zde se jich vyskytuje více. Dokonce zde byl portován i onen *Metamod*, který byl potřebný u předcházející verze.



Obrázek 3.4: Typický případ mnoha zanořených pluginů

³Aplikační programové rozhraní

3.3 Id tech engine

Tento engine využívá v různě modifikovaných podobách nejvíce her na světě. Engine samotný v jeho originální verzi doznal od své první podoby již několik vylepšení a nových verzí. Využívá jej velká spousta herních titulů, které jsou mezi hráčskou komunitou velice oblíbené a úspěšné.

U drtivé většiny dedikovaných serverů využívajících tento engine, případně jeho modifikované verze, je dostupná pouze verze pro operační systém windows, jenž vytváří samotní autoři této hry. Verze pro operační systém linux samotní autoři hry nevytvářejí a nechávají to často na externích firmách, případně jednotlivcích. V případě všech níže zmíněných her se o portování dedikovaného serveru pro operační systém linux doposud staral jeden člověk, takže všechny tyto dedikované servery dodržují stejný koncept. Nevýhoda tohoto řešení z hlediska koncových uživatelů je bohužel v časové dostupnosti, kdy doba od vydání verze pro systém Windows a pro systém Linux je řádově od jednoho do deseti týdnů. To je problém především těsně po vydání hry, kdy se začínají objevovat nové a nové chyby a často tedy vycházejí opravné aktualizace. Při každé nově vydané verzi dedikovaného serveru je poté nutno čekat několik dnů až týdnů, než bude vytvořena také verze pro operační systém Linux. Z tohoto důvodu vznikají situace, kdy jedna část serverů na internetu běží na novější verzi a druhá část na verzi zastaralé. Klient se s jinou verzí, než která je na serveru, samozřejmě nemůže do hry připojit. Pro tento typ serverů neexistuje žádný nástroj pro automatickou aktualizaci serveru. Je tedy nutné toto provádět manuálně. Výhoda těchto typů serverů však spočívá v tom, že všechny soubory, které server pro svou funkčnost vyžaduje, jsou komprimovány do několika větších archivů, takže orientace v souborovém systému tohoto serveru je oproti jiným značně jednodušší. Mezi hry postavené na tomto engine patří:

- Call of Duty
- Call of Duty: United Offensive
- Call of Duty 2
- Call of Duty 4
- Call of Duty: World at War
- Quake 3
- Return to Castle Wolfenstein
- Medal of Honor

3.3.1 Query

Tento protokol je opět implementován na síťovém protokolu UDP/IP. V prvním kroku je třeba serveru zaslat následující řetězec obsahující tyto znaky:

```
\xFF\xFF\xFF\xFFgetstatus\x00
```

Jako odpověď poté dostaneme paket s hlavičkou obsahující

```
\xFF\xFF\xFF\xFF
```

a tělem paketu obsahujícím samotné informace o serveru. Jako první se v odpovědi vyskytuje řetězec:

```
statusResponse\n
```

Parametry serveru jsou poté ve tvaru parametr a hodnota, oddělené zpětným lomítkem. Jednotlivé páry parametrů jsou poté opět rozděleny zpětným lomítkem.

```
statusResponse
\_GV_streaming\enabled - v5.2b\g_antilag\1\g_gametype\sd\gamename\
Call of Duty 2\mapname\ mp_carentan\protocol\118\scr_friendlyfire
\1\scr_killcam\0\shortversion\1.3\sv_allowAnonymous\0\
```

3.3.2 Rcon

Jelikož tento engine má stejné kořeny jako engine využitý ve hrách *Half-Life*, implementace rcon protokolu je v něm stejná. Rozbor tohoto protokolu tedy nalezneme v kapitole 3.1.2 na straně 8.

3.3.3 Pluginy

Pro tento typ her opět existuje celá řada pluginů a modifikací. Jejich instalace je oproti předchozím případům často přímočařejší a jednodušší. Jak již bylo řečeno výše, všechny datové soubory, která hra využívá jsou komprimovány do několika větších balíčků. Tyto soubory jsou uloženy v jednom hlavním adresáři s názvem *main*. Do tohoto adresáře se taktéž nahrávají pluginy, které se stejně jako všechny ostatní soubory s příponou *.iwd*, *.pk3* načítají a spouštějí automaticky po startu serveru. Na problém však narazíme, pokud na server nahrajeme více modifikací, které však navzájem kolidují a nemohou být spuštěny současně. Jednoduchým řešením by tedy bylo smazat danou modifikaci, kterou zrovna nechceme používat, aby hra fungovala vpořádku. Autoři tohoto engine nám však předložili elegantnější řešení v podobě využití spouštěcího parametru s názvem *fs_game*. Uvedeme-li jako hodnotu tohoto parametru název nově vytvořeného adresáře v kořenovém adresáři herního serveru, připojí se její obsah automaticky do hlavního adresáře *main* serveru. Pokud tedy používáme více modifikací, které často měníme, je vhodné toto upřednostnit a vytvořit adresář zvlášť pro každý plugin před věčným nahráváním a mazáním modifikací ze serveru.

3.4 Série her Battlefield

Do této kategorie her spadají všechny tituly Battlefield od společnosti DICE. Tato série her se oproti většině ostatních liší především ve dvojí verzi dedikovaných serverů. Hra samotná totiž podporuje statisky a hodnoty každého hráče, které si sami hráči během hraní postupně vylepšují. Tyto statistiky se uchovávají na centrálním serveru zajištěným samotným autorem hry. Servery, které tuto funkci podporují, se oficiálně nazývají *Ranked* servery⁴. Tyto servery nejsou volně dostupné ke stažení a pro jejich provoz je nutné zakoupit licenci, která je poměrně nákladná a je proto lepší zakoupit takový server u některého poskytovatele, který jej má v nabídce. Tyto servery dále podléhají pravidlům, které mají zajistit, aby statistiky hráčů nemohly být podvodným jednáním upravovány (například pomocí změny skriptů na serveru, které tyto statistiky počítají). Všechny skripty na serveru jsou psány v jazyce *Python*, což případné podvody jen ulehčuje. K zamezení těchto činností slouží pravidla, která musí dodržovat každý poskytovatel, který licenci na provoz získal. Mezi nejdůležitější pravidlo patří zákaz přístupu k souborům serveru například pomocí FTP. Plné znění pravidel lze nalézt na této adrese <http://wiki.bf2s.com/other/roe#1-globally-enforced-ranked-server-rules>. Všechny tyto servery mají samozřejmě

⁴můžeme volně přeložit jako „server s hodnocením“

možnost být spuštěny standardně bez systému hodnocení, u nichž je již možnost případné libovolné modifikace hry a nevztahuje se na ně žádný licenční poplatek.

Aby bylo vůbec možno tyto servery nějakým způsobem ovládat, vzhledem k tomu, že je k nim zakázán přístup přes FTP, vznikla k těmto účelům aplikace nazvaná *bf2cc*⁵. Ta se skládá ze dvou částí a to z části klientské a části serverové. Každá níže vyjmenovaná hra má svou vlastní verzi této aplikace, jejíž název odpovídá zkratce dané hry s přidanou koncovkou „cc“. Tento program umožňuje měnit veškeré parametry serveru pomocí příjemného grafického rozhraní, které ulehčí proces správy serveru a jeho nastavení. Nevýhoda tohoto řešení spočívá ve skutečnosti, že program je vyvíjen v *.NET Framework* od společnosti Microsoft. V operačním systému Linux je tedy nutností využití volně dostupného prostředí *mono* a to pouze ve verzi 1.1.12. Žádná jiná verze není s touto aplikací kompatibilní. Vzhledem k výše zmíněným skutečnostem jsem se rozhodl v této práci neimplementovat tuto skupinu her. Mezi tyto hry patří především následující tituly:

- Battlefield 1942
- Battlefield 2
- Battlefield 2142
- Battlefield: Bad company 2

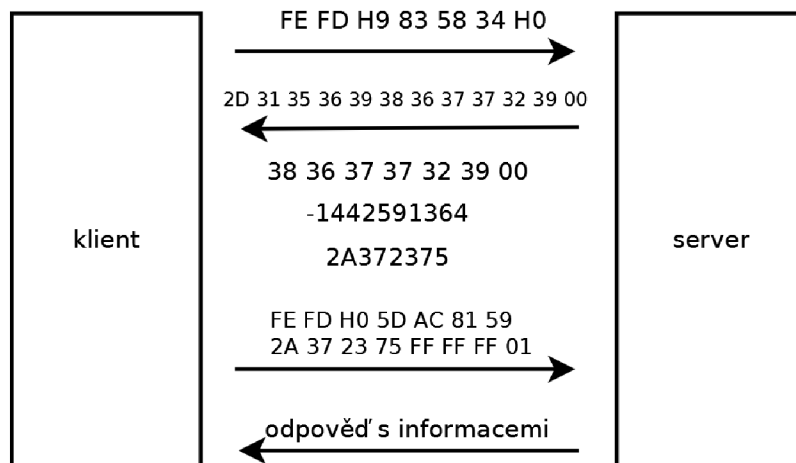
3.4.1 Query

Protokol je implementován na síťovém protokolu UDP/IP. Port, na kterém server naslouchá požadavkům se narozdíl od her uvedených výše liší. Tyto hry mají herní a query port rozdílný. Z informací query paketu je poté možno zjistit port pro připojení do hry, na který se klient může připojit. Podle [2] server podporuje odlehčenou a plnou verzi dotazování. V externích aplikacích se běžně používá plná verze dotazu. V prohlížeči serveru se poté používá odlehčená verze, která zasílá méně dat, aby při aktualizaci seznamu nedocházelo k zbytečně velkému síťovému provozu. Protokol má v sobě implementován částečnou ochranu proti DoS⁶ útokům. Komunikace tedy probíhá následovně:

- Zaslání řetězce `\xFE\xFD\xH9\x83\x58\x34\xH0` na server.
- Server odpoví zprávou obsahující náhodný řetězec.
- Z tohoto řetězce odstraníme prvních 5 bajtů a převedeme na číslo datového typu *integer*.
- Vzniklé číslo převedeme opět na řetězec, z něhož odstraníme posledních 8 bajtů.
- Řetězec, který tímto vznikl (4 bajty), vložíme do předem určeného řetězce `\xFE\xFD\xH0\x5D\xAC\x81\x59-----\xFF\xFF\xFF\x01`.
- Výslednou posloupnost znaků odešleme na server, výsledkem bude seznam parametrů oddělený zpětnými lomítky podobně jako v kapitole 3.3.1.

⁵<http://www.bf2cc.com/>

⁶DoS - Denial of Service, znepřístupnění služby



Obrázek 3.5: Příklad komunikace

3.4.2 Rcon

Tento typ her plně podporuje rcon protokol, jenž je implementován na síťovém protokolu TCP/IP. Jako jediná ze všech her zde uvedených podporuje šifrování rcon hesla. Po navázání spojení se serverem je klientovi zaslána uvítací zpráva ve tvaru:

```

### Battlefield 2 default RCON/admin ready.
### Digest seed: náhodný_řetězec
  
```

Z této zprávy je potřeba vyseparovat náhodný řetězec, s kterým se dále pracuje. V dalším kroku je potřeba se autentizovat. To provedeme zasláním speciálního řetězce, který získáme spojením výše zmíněného náhodného řetězce přijatého ze serveru v minulém kroku s rcon heslem. Nově vzniklý řetězec je poté nutno zašifrovat algoritmem MD5 a odeslat serveru ve tvaru

```
login šifrované_heslo
```

Vrátí-li se ze serveru odpověď ve tvaru `Authentication successful, rcon ready..`, můžeme začít posílat serveru příkazy v tomto formátu

```
exec příkaz.
```

V této podkapitole jsem čerpal z [4].

3.4.3 Pluginy

Pro tuto hru existuje několik modifikací, které však svou velikostí často přesahují desítky či stovky megabajtů. Tyto modifikace je zapotřebí nahrát každému klientovi do hry, aby se mohl připojit na takto upravený server. Nejde tedy přímo o pluginy pro server v pravém slova smyslu a proto se implementací správy pluginů pro tento typ her nebudeme nadále v této práci zabývat.

3.5 Unreal engine

Od roku 1998, kdy byla vydána první verze tohoto engine, uběhla již řada let a tento jeden z nejznámějších engine je nyní již ve verzi 3. Mezi stále hojně používané však pořád patří verze 2.0 až

2.5, na kterých staví převážná část herních titulů, o kterých se zde budeme zmiňovat. Dostupnost pro operační systém linux bývá standardem, avšak opět se zpožděním oproti verzi pro operační systémy Windows. Hlavní nevýhoda v tomto případě tkví v extrémní neřehlednosti konfiguračního souboru serveru. Ten obsahuje přes 1000 řádků konfiguračních proměnných a pro neznalého člověka je problém v něm cokoli najít. Server pro tuto hru nepodporuje žádnou formu automatických aktualizací, a je proto třeba udržovat server aktualizovaný manuálně. Mezi hry, které byly vytvořeny na tomto engine patří:

- Unreal Tournament 2003
- Unreal Tournament 2004
- Americas Army 2
- Unreal 3

Tyto informace jsem čerpal z [7].

3.5.1 Query

Podobně jako u série her Battlefield, i tento engine používá pro dotazování a pro samotné připojení do hry jiný port. Protokol je opět implementován na síťovém protokolu UDP/IP. Pro získání základních informací ohledně serveru (název serveru, port pro připojení do hry, počet hráčů atd.) je třeba zaslat řetězec obsahující `\info\`. Odpověď serveru na tento požadavek je ve tvaru `\parametr\hodnota\n`. Pro detailnější popis stavu serveru je možno odeslat příkaz `\basic\info\rules\players\`.

3.5.2 Rcon

Protokol rcon pro tuto hru bohužel není dostupný. V samotné hře je sice implementována podpora pro ovládání serveru příkazy, ta však nejde použít externě mimo samotnou hru. Toto znemožňuje jakékoliv změny parametrů pomocí externích programů. Aby bylo možno server externě nějakým způsobem spravovat, je možno spustit paralelně se samotným serverem i server webový, implementovaný přímo v herním serveru samotném. Přes tuto webovou stránku je poté možno server spravovat.

3.5.3 Pluginy

Pro tyto hry existuje mnohem méně pluginů než u předcházejících her. Často se však jedná o větší modifikace v rozsahu jednotek až stovek megabajtů, což není ideální z hlediska zdlouhavého nahrávání takto velkých modifikací na server.

3.6 Série Trackmania

Tato hra jako jediná ze všech v této práci zmíněných je založena na herním žánru arkádového simulátoru aut. První díl této hry byl vydán v roce 2006. Hra je založena na klient-server architektuře. Hra je taktéž výjimečná tím, že je jako jediná zde uvedená zcela zdarma. Pro podporu autorů této hry lze však zakoupit i placenou verzi, která nabízí několik výhod oproti verzi, která je volně stažitelná z internetu. Dedikovaný server pro tuto hru byl vydán pro oba hlavní operační systémy používané pro hostování serverů tj. Linux a Windows. Pro správnou funkci tohoto herního serveru je třeba mít

vytvořený herní účet. Ten se vytváří přímo ve hře samotné, tudíž pro spuštění serveru je potřeba mít nainstalovanou a sprovozněnou samotnou hru. Tento titul se bohužel vyznačuje složitou konfigurací. Do této série spadají tyto herní tituly

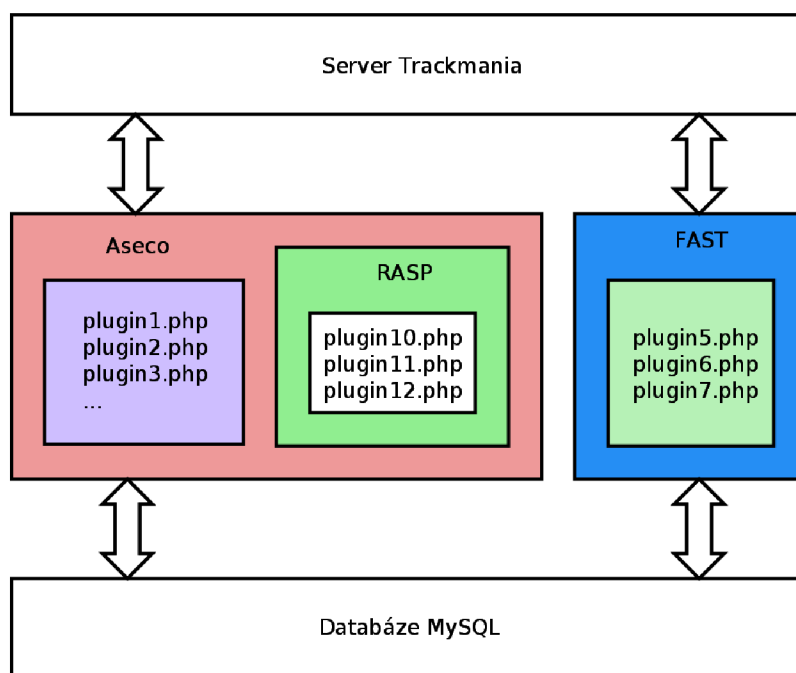
- Trackmania Nations,
- Trackmania Forever.

3.6.1 Query a ovládání

Tato hra je výjimečná použitím protokolu *XML-RPC* pro veškerou komunikaci se serverem. Pomocí tohoto protokolu můžeme ze serveru získat informace jako například počet hráčů, aktuální mapu a název serveru. Pro ovládání serveru je zapotřebí znát administrátorské heslo, bez kterého není možné tímto protokolem provádět pokročilejší akce.

3.6.2 Pluginy

Pro tuto hru existuje řada pluginů a několik *API* pro jejich snazší tvorbu. Všechny tyto pluginy jsou napsány v programovacím jazyce php a komunikují se serverem právě pomocí protokolu *XML-RPC*. Tyto pluginy často ke svému běhu vyžadují připojení k databázi, což znesnadňuje proces zautomatizování instalace těchto rozšíření. Mezi nejznámější pluginy patří například *Aseco*, které zároveň plní funkci frameworku pro tvorbu dalších pluginů. Nevýhoda tohoto řešení spočívá v tom, že tyto pluginy, komunikující se serverem skrze protokol *XML-RPC*, je nutno externě spouštět. Bez přístupu ke konzoli počítače tedy není možno tento plugin spustit. Běžný server s nainstalovanými základními pluginy může mít strukturu jako na obrázku 3.6.



Obrázek 3.6: Struktura pluginů hry Trackmania

Kapitola 4

Implementace

Pro implementaci tohoto projektu jsem zvolil skriptovací jazyk Ruby. Důležitým faktorem pro výběr právě tohoto objektově orientovaného jazyka je jeho silná vyjadřovací schopnost a snadná práce s textem, která je v tomto případě zapotřebí. Pro tento jazyk existuje pracovní prostředí pro tvorbu webových aplikací nazvané *Ruby on Rails*, ve kterém byl celý program ve formě webové aplikace napsán. *Ruby on Rails* je založeno na moderní architektuře *MVC* (model-view-controller), která se snaží o maximální oddělení části logické, datové a uživatelského rozhraní. Forma webové aplikace byla zvolena kvůli jednoduché přenositelnosti na veškeré operační systémy včetně mobilních zařízení.

4.1 Databázový návrh a implementace

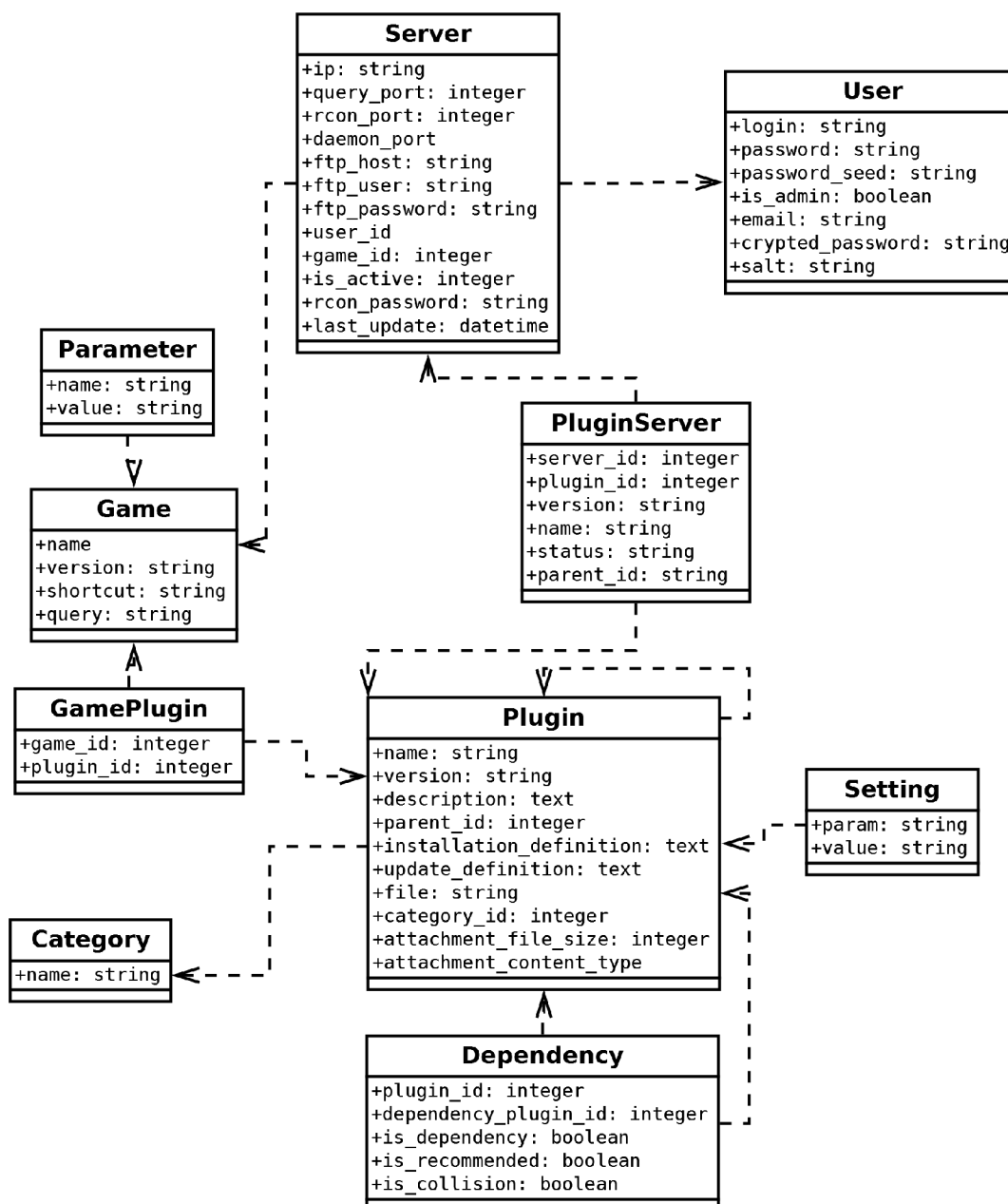
Ruby on Rails (dále již jen Rails) umožňuje práci se všemi nejpoužívanějšími typy databází. V této práci však využijeme databáze s otevřeným zdrojovým kódem *MySQL*. Rails vyžaduje podporu transakcí, je tedy nutné zvolit typ tabulek v databázi jako *InnoDB*. Pro usnadnění práce s vytvářením struktury databáze podporují Rails tzv. migrace. Ty slouží podobně jako systémy pro správu verzí projektů, avšak pro strukturu databáze. Každá verze poté obsahuje definici toho, co je potřeba v databázi změnit, abychom se dostali na verzi vyšší popřípadě nižší. Jako příklad může posloužit vytvoření tabulky uživatelů:

```
class CreateUsers < ActiveRecord::Migration
  def self.up
    create_table :users do |t|
      t.string :login
      t.string :password
      t.boolean :is_admin

      t.timestamps
    end
  end

  def self.down
    drop_table :users
  end
end
```

Tato konstrukce zajistí vytvoření požadované tabulky a její naplnění sloupci s danými datovými typy. Při práci s databází se k ní nepřístupuje přímo pomocí *SQL* dotazů, avšak pomocí architektonického návrhu *Active Record* pro práci se zdroji dat. Toto zapouzdření umožňuje v Rails použití jakékoliv jiné databáze, která je podporována, bez zásahu do zdrojového kódu programu. Návrh databázového schéma je zobrazen na obrázku 4.1.



Obrázek 4.1: Databázové schéma tříd

Tento návrh plně zohledňuje požadavky plynoucí ze složité struktury pluginů některých her. Je potřeba přesně definovat, které pluginy jsou závislé na jiných. Je tedy potřeba vytvořit stromovou strukturu objektu *Plugin*. Toho jednoduše docílíme přidáním atributu *parent_id* tomuto objektu. Ten odkazuje na předka, který se taktéž může dále odkázat na libovolného předka. Toto využijeme

především u her založených na *GoldSource* enginu dle obrázku 3.2 na straně 10. Další podstatnou částí databázového návrhu je možnost využití závislostí na jiných pluginech. Toho je docíleno vazební tabulkou s parametry

- závislost,
- doporučení,
- kolize.

Závislost určuje, že plugin pro svůj běh vyžaduje jiný plugin, který musí být taktéž nainstalován. Doporučení určuje pluginy, které je vhodné nainstalovat. Nebudou však instalovány automaticky jako je tomu v předchozím případě. Poslední možností je kolize. Zjistí-li systém, že je na serveru nainstalován některý plugin, který nemůže současně fungovat s pluginem, který chceme nainstalovat, bude instalace předčasně ukončena.

4.2 Hlavní plugin

Rails podporují možnost vytvoření vlastních aplikačních pluginů pomocí tzv. generátorů. Tyto generátory umí vygenerovat kostru tohoto pluginu tak, aby odpovídal normám. Pro vygenerování této kostry stačí v domovském adresáři projektu zadat v příkazové řádce příkaz `script/generate plugin název_pluginu`. Kostra pluginu je vytvořena v adresáři `vendor/plugins`. Všechny pluginy, které v tomto adresáři jsou, se automaticky načítají, dodržíme-li správnou konvenci. Plugin, který byl pro tento účel vytvořen, obsahuje jádro celé aplikace a je rozdělen do těchto tří modulů:

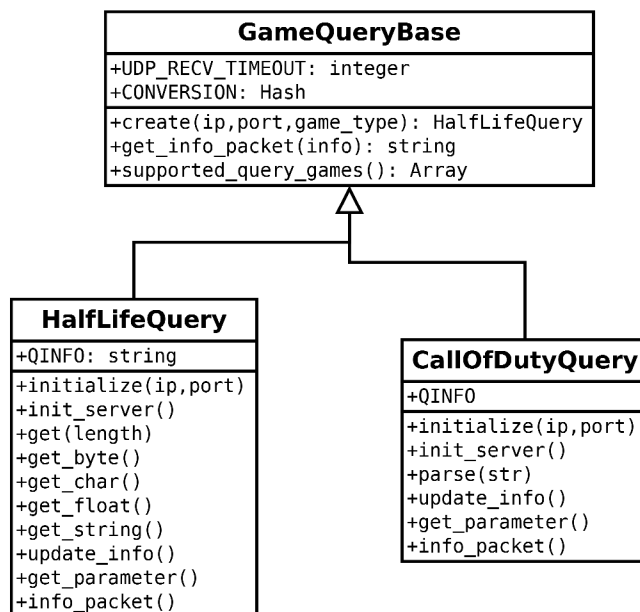
- implementace dotazovacích protokolů query,
- implementace vzdáleného ovládání rcon,
- implementace systému pro detekci pluginů na serveru.

Oddělení této části aplikace do samostatného pluginu umožní budoucí jednodušší a přehlednější rozšiřování aplikace o další typy her a možnosti ovládání.

4.2.1 Dotazovací protokoly herních serverů

V tomto modulu jsou implementovány query protokoly herních serverů. Jeho základ tvoří třída nazvaná *GameQuery*. V aplikaci tuto třídu voláme metodou `create` s parametry ip adresy, portu a typu hry. Typy her, které tato třída podporuje jsou definovány v konstantě `CONVERSION`. Herní typy zadáváme jako zkratku, která se poté konvertuje na plný název hry (pouze však malá písmena a podtržítka). Metoda vrátí na základě typu hry správnou instanci objektu, ve kterém je implementován požadovaný query protokol dané hry. V tomto objektu je vyžadována implementace následujících metod:

- `update_info()` sloužící pro obnovení dat
- `get_parameter(parametr)` pro přečtení parametru
- `initialize` s parametry ip adresy a portu

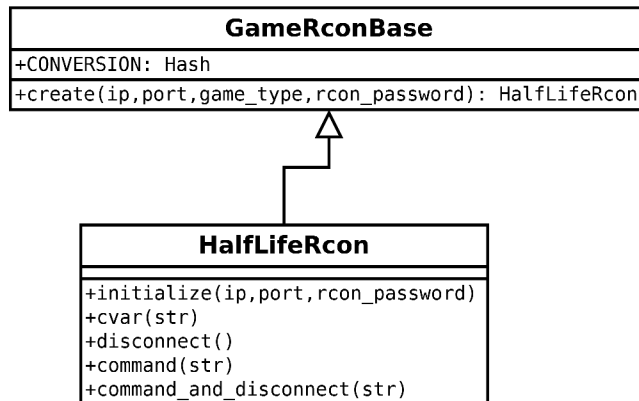


Obrázek 4.2: Modul dotazovacích protokolů

4.2.2 Vzdálené ovládání pomocí rcon protokolu

Modul vzdáleného ovládání implementuje protokoly rcon pro zasílání příkazů hernímu serveru. Základem tohoto modulu *GameRcon* je třída nazvaná *GameRconBase*. Její funkcí je pouze pomocí třídní metody *create* vrátit správný objekt obsahující implementaci rcon protokolu dané hry. Je zde opět využita konstanta *CONVERSION*, definovaná jako asociativní pole, pro konverzi zkratk her na plné názvy, která taktéž pomocí třídní metody *supported_games* zajišťuje vypsání implementovaných her. Metoda *create* vrátí na základě typu hry instanci objektu, ve kterém je implementován požadovaný rcon protokol dané hry. V tomto objektu musí být implementovány minimálně následující metody

- *command(příkaz)*, která vrátí odpověď serveru na zadaný příkaz,
- *disconnect* pro ukončení práce, případně uzavření otevřených soketů,
- *command_and_disconnect(příkaz)* pro zaslání příkazu a okamžitého ukončení práce,
- *initialize* s parametry ip adresy, portu a rcon hesla.



Obrázek 4.3: Modul vzdáleného ovládání pomocí rcon protokolu

Tento modul využívá pluginu *rcon*¹ vytvořený pro jazyk Ruby. Ten podporuje rcon protokoly pro *Source* a *GoldSource* engine. V tomto pluginu je bohužel chyba u přijímání paketů v odpovědích. Projevuje se u zmíněného *GoldSource* engine, kdy je vždy přijat pouze první paket, i když byla odpověď rozdělena do více paketů. Řešením je tedy přepsat metodu pro přijímání paketů. K tomu se přímo nabízí dynamičnost, kterou ruby disponuje. Stačí tedy danou metodu přímo v aplikaci dynamicky předefinovat následujícím kódem.

```

class RCon::Query::Original
  def retrieve_socket_data
    return "" if @socket.nil?

    retval = ""
    while IO.select([@socket], nil, nil, 1) != nil
      packet = @socket.recv(1440)
      packet.sub! /\xFF\xFF\xFF\xFF#{@server_type}/, ""
      packet.sub! /\x00+$/, ""
      retval << packet
    end

    return retval
  end
end

```

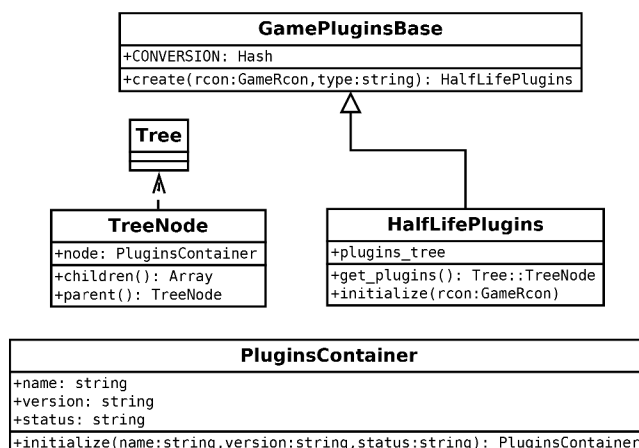
4.2.3 Detekce pluginů na serveru

Pro detekci spuštěných pluginů na serveru byl vytvořen tento modul. Využívá dostupné metody ovládání serveru ke zjištění pluginů, které jsou spuštěny. Modul se jmenuje *GamePlugins* a jsou v něm implementovány 2 třídy. K své funkci využívá třídu *Tree*, kterou je nutné vyžádat příkazem `require 'tree'`. Třída *PluginsContainer* slouží jako kontejner k uložení jednoho pluginu. O něm jsou uchovávány informace ohledně názvu, verze, a stavu, v jakém se nachází, je-li tato informace dostupná. Třída *GamePluginsBase* je základní třídou, jenž slouží k výběru správného parseru pluginů podle zadaného typu hry. Metoda *create* této třídy vrátí instanci třídy, ve kterém je

¹ke stažení na <http://rubyforge.org/projects/rcon/>

implementována detekce pluginů nahraných na serveru. Pro hru *Half-Life* tedy vrátí instanci třídy *HalfLifePlugins*, která dědí od třídy *GamePluginsBase*. Konstruktor této třídy, metoda *initialize* přijímá jako parametry objekty *GameRcon*, *FTP* a *GSClient*. Tyto objekty poté využívá k přístupu na server, odkud zjistí požadované informace o spuštěných pluginech.

Pluginy jsou uloženy ve formě stromu. Každý uzel stromu je tvořen třídou *PluginsContainer*, která o něm uchovává informace. Jelikož tato operace může být časově náročná, provádí se obnovení detekovaných pluginů pouze na žádost uživatele po kliknutí na příslušnou akci.



Obrázek 4.4: Modul detekce pluginů na serveru

4.2.4 Instalace pluginů

Instalace pluginů na herní servery nebývá nijak standardizována. Některé pluginy mohou obsahovat pouze jeden soubor, který stačí nahrát do předem určeného adresáře na serveru. Problém nastává v případě, kdy se plugin skládá z více souborů, případně je potřeba změnit konfigurační soubory serveru. Toto činí mnoha nezkušeným administrátorům problém. Pro zautomatizování tohoto procesu je proto potřeba vytvořit systém, který umožní u každého pluginu definovat postup jeho instalace. Při samotné instalaci tedy systém projde celou definicí příkazů a postupně je provede. Problém může nastat, pokud se postup instalace liší pro různé operační systémy, na kterém server běží. Je-li instalace pluginů stejná pro operační systém windows i linux, zapíšeme do konfiguračního souboru pouze příkazy nutné k instalaci. V opačném případě před samotnou definicí příkazů uvedeme na samostatný řádek název operačního systému, pro který je blok příkazu relevantní. Tuto definici je potřeba napsat pro instalaci, aktualizaci a případně smazání. K dispozici jsou následující příkazy:

- `full_copy` - zkopíruje všechny soubory, automaticky přepisuje již existující soubory
- `copy <zdroj> <cíl>` - zkopíruje soubor, automaticky přepisuje již existující soubory
- `remove <soubor>` - smaže soubor
- `mkdir <adresář>` - vytvoří nový adresář
- `mkfile <soubor>` - vytvoří nový soubor
- `set_cvar <konfigurační_soubor> <parametr> <hodnota>` - nastaví daný parametr na novou hodnotu ve specifikovaném konfiguračním souboru

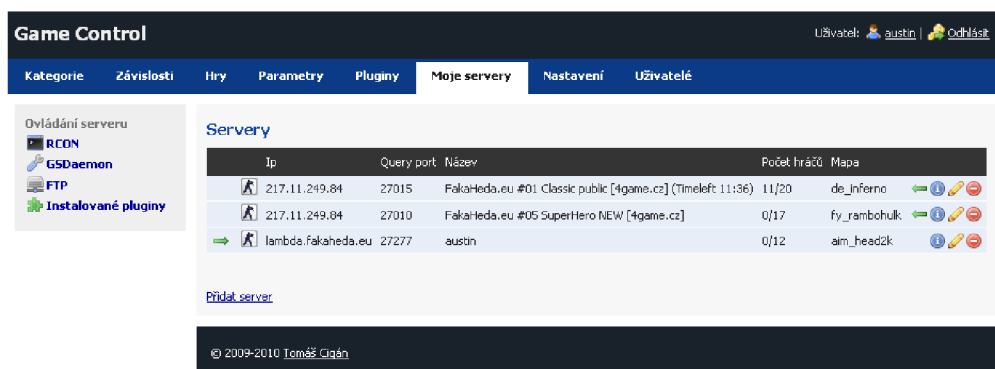
- `comment_line <konfigurační_soubor> <vzor>` - zakomentuje řádek podle zadaného vzoru
- `remove_line <konfigurační_soubor> <vzor>` - odstraní řádek odpovídající vzoru
- `add_line <konfigurační_soubor> <řetězec>` - přidá řádek do konfiguračního souboru
- `restart_server` - restartuje server
- `stop_server` - vypne server
- `start_server` - zapne server
- windows
- linux
- general

Plugins jsou v systému uloženy jako jediný soubor komprimovaný do archivu zip. Při instalaci se tento soubor dekomprimuje do dočasné složky, provede se instalace na server a poté se opět smaže, aby nezabíral místo na pevném disku. Jako příklad nejjednodušší instalace pluginu, kde pouze zkopírujeme plugin, připíšeme jej do konfiguračního souboru a poté restartujeme server, může vypadat takto:

```
copy plugin.amxx /cstrike/addons/amxmodx/plugins/
add_line /cstrike/addons/amxmodx/configs/plugins.ini plugin.amxx
restart_server
```

K instalaci systém využije dostupných metod komunikace se serverem. Je-li dostupné pouze připojení přes FTP, není možno provádět některé operace podle instalační definice. Operace pro restartování, zapínání a vypínání serveru je možno provést pouze pokud je dostupné připojení k GS-Daemon serveru. Při instalaci pluginu nejprve systém zkontroluje závislosti na jiných pluginích a postupně je všechny nainstaluje.

4.3 Uživatelské rozhraní



Obrázek 4.5: Grafické rozhraní

Uživatelské rozhraní je rozděleno do tří základních bloků. První blok je standardní horní menu. Do tohoto menu má ke většině záložek přístup pouze administrátor. Běžný uživatel vidí pouze záložku „Moje servery“, na které se nachází seznam vlastních serverů přidaných do systému a „Pluginy“. Druhým blokem je levé menu. Položky tohoto menu se vážou na právě aktivní server, který je nutno vybrat v seznamu serverů a označit jako aktivní. V systému není možno pracovat s více servery současně. Posledním blokem je samotný obsah stránky. Administrátor je v systému odlišen od normálních uživatelů databázovým atributem *is_admin*. Přihlašuje se stejně jako běžný uživatel. Není tedy nutné speciální přihlašování pro vstup do administrace. Administrátor může změnou atributu *is_admin* v seznamu uživatelů přidávat či odebírat další administrátory.

4.3.1 Seznam serverů

K tomu, aby mohl uživatel se serverem pracovat, je potřeba, aby jej nejprve přidal do svého seznamu. Aby bylo možné přidat nový server do seznamu, je potřeba zadat o něm následující údaje:

- typ hry
- IP adresa
- query port
- rcon port
- rcon heslo
- GSDaemon port, login, heslo
- IP adresa FTP serveru
- FTP uživatel
- FTP heslo

4.3.2 Přidávání pluginů do systému

V systému bude možno definovat stromovou strukturu pluginů a závislosti mezi pluginy. Administrátor musí přesně definovat, je-li plugin potomkem některého již v systému vytvořeného pluginu. Pokud plugin vyžaduje pro svůj chod instalaci některého dalšího, je třeba tyto závislosti explicitně definovat. Ke každému pluginu je potřeba dopsat definici instalačního, a aktualizacího skriptu, verzi, název a přidat samotný plugin ve formě zip archívu. Pluginy je možno řadit do kategorií pro jednodušší vyhledávání.

4.3.3 Ovládání serveru

K veškerému ovládání serveru slouží levý postranní blok. V tomto bloku se nachází jednotlivé pomůcky, pomocí kterých lze server ovládat. Pod záložkami Rcon a GSDaemon se nachází režim vzdálené konzole. Do příkazového řádku je možno psát požadované příkazy, které server zpracuje a odpověď vrátí zpět. Konzole je implementována pomocí technologie *AJAX*, tudíž se stránka během odeslání požadavku neaktualizuje a výstup konzole se po každém požadavku připojí na konec předchozí odpovědi. V tomto bloku se nachází následující položky:

- Rcon

- GSDaemon
- FTP
- Parametry serveru
- Instalované pluginy

4.3.4 Instalace a aktualizace pluginů

Instalaci pluginů provádíme pod záložkou „Pluginy“, kde si vyhledáme požadovaný plugin, který chceme nainstalovat. Aktualizace pluginů se poté provádí opět z levého ovládacího menu, pod odkazem „Instalované pluginy“. Na této stránce se zobrazí všechny detekované pluginy na serveru. Pokud byl plugin v systému nalezen, označí se jako spárovaný pomocí zelené ikony a je možné jej aktualizovat, je-li k dispozici novější verze.

4.4 Bezpečnost

4.4.1 Autentizace

Jako systém pro autentizaci a autorizaci jsem zvolil plugin pro Ruby on Rails zvaný *restful_authentication*. Ten zajišťuje kvalitní a především bezpečný způsob jak samotného přihlašování, tak i způsobu uložení hesel v systému. Hesla jsou v databázi šifrována pomocí SHA-1 s pomocí tzv. semínka a klíče který je jedinečný pro celou aplikaci.

4.4.2 Autorizace

V systému existují tři role, do kterých se uživatel může dostat. Mezi ně patří

- nepřihlášený,
- přihlášený běžný uživatel,
- přihlášený administrátor.

Pokud uživatel není přihlášený, nemůže v systému dělat nic jiného, než se registrovat. Všechny akce v kontrolérech mají nastavenou tzv. *before_filter* metodu, která zajistí, že před každým požadavkem bude zkontrolováno, je-li uživatel autorizován k dané akci. Administrátoři jsou v systému rozlišeni od běžných uživatelů pouze pomocí databázového atributu. Není tedy možné přidělovat různá práva různým rolím, či uživatelům.

Kapitola 5

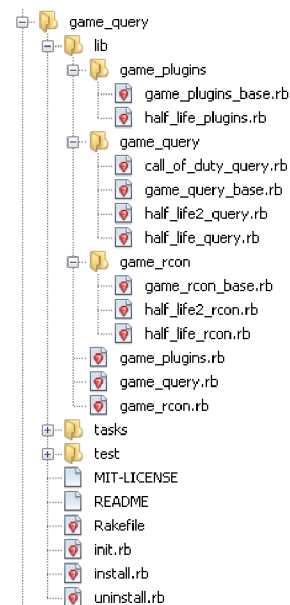
Demonstrace rozšíření podpory her

5.1 Query modul

Pokud budeme chtít aplikaci rozšířit o podporu dalších herních serverů, stačí k tomu upravit moduly hlavního pluginu. Stěžejní je pro každou hru přidání podpory minimálně pro protokol query. K tomu stačí vytvořit soubor s příponou `.rb` do adresáře pluginu nazvaného `game_query`, který je umístěn v adresáři `vendor/plugins` projektu. Soubor je nutné pojmenovat dle názvu hry a připojit slovo `query`. Místo mezer je samozřejmě nutno použít podtržítka. V samotném souboru je poté nutno uvést, do kterého modulu třída, kterou později nadefinujeme, spadá.

```
module GameQuery
  class NovaHraQuery < GameQueryBase
    def initialize(ip, port)
    end
    def get_parameter(param)
    end
    def update_info
    end
  end
end
```

V této nově vzniklé třídě je tedy nutno nadefinovat minimálně metody `get_parameter` a `update_info` dle kapitoly 4.2.1 na straně 20. Do souboru `game_query_base.rb` je poté nutno uvést správnou konverzi zkratky hry na název souboru, který jsme vytvořili. Pro tento příklad by tedy konverze mohla vypadat takto `'nh' => 'nova_hra'`. Přidání podpory tímto modulem však zajistíme pouze to, že nová hra půjde do systému přidat a server bude viditelný v seznamu uživatelových serverů.



Obrázek 5.1: Adresářová struktura hlavního pluginu

5.2 Rcon modul

Abychom mohli se serverem nějak manipulovat, je potřeba přidat také podporu rcon protokolu. To zajistíme velice podobným způsobem. V adresáři `game_rcon` vytvoříme nový soubor s názvem hry a připojeným slovem `rcon`. V něm poté uvedeme definici správného modulu a vytvoříme novou

třidu s názvem stejným jako název souboru avšak písmena za podtržítkem nahradíme za velká a samotné podtržítka smažeme. Kostra tohoto souboru by tedy měla vypadat následovně:

```
module GameRcon
  class NovaHraRcon
    def initialize(ip, port, rcon_password)
    end
    def cvar(tmp)
    end
    def disconnect
    end
    def command(tmp)
    end
    def command_and_disconnect(tmp)
    end
  end
end
```

5.3 Modul detekce pluginů

Poslední částí je podpora instalace a detekce spuštěných pluginů na serveru. Z hlediska implementace je tato část nejnáročnější, jelikož data, která chceme získat, tj. seznam spuštěných pluginů na serveru, musíme nějakým způsobem vydolovat. K dispozici tomuto modulu je tedy přístup k souborům herního serveru přes FTP, rcon protokol a případně GSDaemon. Ve většině případů si vystačíme s použitím rcon protokolu jehož výstup je nutno parsovat pomocí regulárních výrazů. Tyto pluginy je poté nutno správně uložit do stromové struktury. S vytvářením souboru, jeho pojmenováním a názvem tříd postupujeme analogicky k předchozím bodům.

```
module GamePlugins
  class NovaHraPlugins < GamePluginsBase
    def initialize(game_rcon, ftp, gsdaemon)
    end
    def get_plugins
    end
  end
end
```

Kapitola 6

Závěr

Cílem této práce bylo nastítnit možnosti zjednodušení správy a ovládání různých typů herních serverů. Hlavním předmětem pak bylo navrhnout a implementovat kompletní flexibilní systém pro správu a automatickou instalaci pluginů pro tyto servery. Mým záměrem nebylo obsáhnout maximum herních serverů, avšak nastítnit možnosti ovládání a správy těch nejpoužívanějších s největším množstvím pluginů.

6.1 Další vývoj

V této práci jsem se především zaměřil na návrh a technickou stránku problému. Aby se mohl tento systém více rozšířit, bylo by zapotřebí věnovat značnou část úsilí také do kvalitního uživatelského rozhraní a flexibilnější možnosti autorizace administrátorů. Jako další část by bylo možné vytvořit databázi konfiguračních proměnných se slovním popisem a tagy, ve kterém by bylo možno jednoduše vyhledávat. Např. chceme-li nastavit omezení kamery ve hře, jednoduše bychom mohli vyhledat, které parametry s touto frází souvisí a nastavit je.

6.2 Zhodnocení odvedené práce

Tato práce mi přinesla mnoho nových poznatků o herních serverech a především o objektově orientovaném programování v jazyce ruby. Taktéž jsem si osvojil návrh databáze a především kaskádových stylů a jazyka html při tvorbě webového rozhraní. Během implementace této práce jsem narazil na problémy především v chybějících dokumentacích k herním serverům při používání protokolů ke komunikaci s nimi.

Literatura

- [1] Faika, J.: *Systém pro správu herních serverů, bakalářská práce*. Brno, FIT VUT v Brně, 2010.
- [2] WWW stránky: BF2142 Query Protocol - BF2 Technical Information Wiki.
URL http://bf2tech.org/index.php/BF2142_Query_Protocol
- [3] WWW stránky: Legacy:UT Server Query.
URL http://en.wikipedia.org/wiki/Quake_engine
- [4] WWW stránky: RCon Protocol - BF2 Technical Information Wiki.
URL http://bf2tech.org/index.php/RCon_Protocol
- [5] WWW stránky: Source queries - Valve Developer Community.
URL http://developer.valvesoftware.com/wiki/Server_queries%
- [6] WWW stránky: Source RCON protocol - Valve Developer Community.
URL
http://developer.valvesoftware.com/wiki/Source_RCON_Pr%otocol
- [7] WWW stránky: Unreal engine - WikiPédie.
URL http://cs.wikipedia.org/wiki/Unreal_Engine#Unreal_Engi%ne_2
- [8] WWW stránky: XML-RPC - WikiPédie.
URL <http://cs.wikipedia.org/wiki/XML-RPC>