

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Gregor Šturm

Razvoj iger z uporabo ogrodja MMORPG

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM
PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

Mentor: doc. dr. Rok Rupnik

Ljubljana, 2018

Copyright. Rezultati diplomske naloge so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavo in koriščenje rezultatov diplomske naloge je potrebno pisno privoljenje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Razvoj računalniških iger je brez posebnih ogrodij precej zahteven in dolgotrajen. TrinityCore je projekt, v okviru katerega nastaja MMORPG kot eno takšnih ogrodij. Proučite ogrodje MMORPG in v okviru tega tudi uporabo emulacij. Na podlagi tega razvijte tudi nekaj praktičnih primerov novih funkcionalnosti za ogrodje.

Zahvaljujem se družini in prijateljem za podporo in spodbude pri pisanju diplomskega dela. Prav tako se zahvaljujem doc. dr. Roku Rupniku za vse nasvete ter strokovno pomoč pri izdelavi diplomske naloge.

Kazalo

Povzetek

Abstract

| | | |
|-------|--|----|
| 1 | Uvod..... | 1 |
| 1.1 | Kaj je TrinityCore..... | 1 |
| 1.2 | Kaj je emulator/emulacija..... | 2 |
| 1.3 | Predstavitev igre World of Warcraft..... | 2 |
| 1.3.1 | Incident »Corrupted Blood«..... | 3 |
| 1.4 | Zakaj je TrinityCore izbral World of Warcraft..... | 4 |
| 1.5 | Moja vloga pri projektu TrinityCore..... | 5 |
| 1.6 | Cilj diplomske naloge..... | 6 |
| 2 | Arhitektura TrinityCore..... | 7 |
| 2.1 | DBC in DB2 za hranjenje statičnih podatkov..... | 7 |
| 2.1.1 | Podatkovna baza DBC..... | 7 |
| 2.1.2 | Podatkovna baza DB2..... | 7 |
| 2.1.3 | Pregled najbolj pomembnih DBC/DB2 datotek..... | 9 |
| 2.2 | Strežnika Auth/Bnet in World..... | 11 |
| 2.2.1 | Strežnik Auth..... | 11 |
| 2.2.2 | Strežnik World..... | 14 |
| 2.3 | Programi za ekstrahiranje podatkov iz igre..... | 19 |
| 2.4 | Distribucija posodobitev..... | 21 |
| 2.4.1 | Posodobitve preko MPQ-datotek..... | 21 |
| 2.4.2 | Posodobitve preko podatkovne baze Hotfixes..... | 23 |
| 2.5 | Podatkovne baze MySQL za strežniške podatke..... | 25 |
| 2.5.1 | Podatkovna baza Auth..... | 25 |

| | | |
|-------|--|----|
| 2.5.2 | Podatkovna baza Characters | 26 |
| 2.5.3 | Podatkovna baza World..... | 27 |
| 3 | Praktični primeri | 29 |
| 3.1 | Implementacija nove funkcionalnosti v TrinityCore strežniku..... | 29 |
| 3.2 | Dodajanje novih objektov v podatkovni bazi..... | 32 |
| 3.2.1 | Tabela creature_template..... | 32 |
| 3.2.2 | Tabela creature_template_outfits | 35 |
| 3.2.3 | Tabela creature_equip_template..... | 36 |
| 3.2.4 | Tabela creature_template_addon..... | 36 |
| 3.2.5 | Tabela creature_text | 37 |
| 3.2.6 | Končen rezultat..... | 38 |
| 3.3 | Izdelava spletne strani za TrinityCore..... | 39 |
| 3.3.1 | Stran za prijavo v sistem..... | 40 |
| 3.3.2 | Uporabnikova nadzorna plošča | 41 |
| 3.3.3 | Stran za upravljanje | 43 |
| 3.3.4 | Stran za ogled zgodovine..... | 44 |
| 4 | Sklepne ugotovitve | 45 |
| | Literatura..... | 47 |

Seznam uporabljenih kratic

| KRATICA | ANGLEŠKO | SLOVENSKO |
|---------------|--|-------------------------------------|
| MMORPG | Massively multiplayer online role-playing game | Masivna večigralska enopotezna igra |
| NPC | Non-playable character | Neigralniški lik |
| GOB | Game Object | Igrški objekt |
| RP | Role-play | Igranje v vlogi |
| ID | Identification / identity | Identifikacija / identiteta |
| PVP | Player versus player | Igralec proti igralcu |

Seznam uporabljenih strokovnih besed

| Angleški izraz | Uveljavljen prevod | Predlagani prevod | V diplomu uporabljen izraz |
|------------------|--------------------|-------------------|----------------------------|
| Quest | Iskanje | Dogodivščina | Dogodivščina |
| Item | Predmet | Predmet | Predmet |
| Game master | / | Upravitelj igre | Upravitelj igre |
| Spell | Čar | Sposobnost | Sposobnost |
| Guild | Ceh | Skupina | Skupina |
| Signature | Podpis | Podpis | Podpis |
| Virtual maps | / | Virtualne mape | Virtualne mape |
| Movement maps | / | Premične mape | Premične mape |
| Maps | Mape | Mape | Mape |
| Experience | Izkušnje | Izkušnje | Izkušnje |
| Battleground | Bojišče | Bojišče | Bojišče |
| Realm | Kraljestvo | Kanal | Kanal |
| Player character | / | Igralniški lik | Igralniški lik |
| Instance | Primer | Instanca | Instanca |
| Fraction | Frakcija | Frakcija | Frakcija |
| Class | Razred | Razred | Razred |
| Race | Rasa | Rasa | Rasa |
| Emulation | Posnemanje | Emulacija | Emulacija |
| Emulator | Posnemovalnik | Emulator | Emulator |
| Expansion | Širitev | Razširitev | Razširitev |

| Angleški izraz | Uveljavljen prevod | Predlagani prevod | V diplomu uporabljen izraz |
|-----------------------|---------------------------|------------------------------|-----------------------------------|
| Level | Raven | Stopnja | Stopnja |
| Admin panel | / | Plošča za upravljanje | Plošča za upravljanje |
| User panel | Uporabniška plošča | Uporabnikova nadzorna plošča | Uporabnikova nadzorna plošča |
| Progression tokens | / | Točke za napredovanje | Točke za napredovanje |

Povzetek

V diplomski nalogi je predstavljen odprtokodni projekt TrinityCore, ki je odlično orodje za učenje tako bodočih programerjev, ki želijo delati v industriji računalniških iger, kot tudi posameznikov, ki jih zgolj zanima, kako le-te delujejo.

Predstavljeni bodo delovanje in koncepti ene izmed najbolj priljubljenih iger žanre MMORPG.

V nalogi si bomo ogledali pomen emulacije (*emulation*) kot poučno orodje, celotno arhitekturo, iz katere sestoji projekt, ter praktične primere, ki prikažejo, kako lahko uporabljamo projekt za pridobitev znanja iz zgradbe in delovanja računalniških iger. Ogledali si bomo tudi praktični primer implementacije novih funkcionalnosti v strežnik, dodajanja novih objektov v podatkovno bazo ter izdelavo specifične spletne strani za upravljanje z igralci. Za konec pa si bomo ogledali še pomanjkljivosti projekta in možnosti za izboljšavo in nadaljnji razvoj.

Ključne besede: TrinityCore, računalniške igre, MMORPG, emulator, poučevanje.

Abstract

The thesis presents an open source project named TrinityCore, which is an excellent framework for the teaching of both future programmers, which are leaning towards working in the industry of computer games, as well as curious individuals interested in how these games work.

The operation and concepts of one of the most popular MMORPG games will be presented. In the thesis, we will be taking a look at the term emulation in the role of an educational tool, the entire architecture of which the project consists and practical examples, which demonstrate how we can utilize the project to obtain knowledge in the way computer games operate and how they are built. We will also be taking a look at an example of the implementation of new functions in the server, adding new objects into the database and the creation of a specific website for managing players. Lastly, we will be taking a look at the problems of the project and ways for improvement and future development.

Keywords: TrinityCore, computer games, MMORPG, emulator, teaching.

1 Uvod

1.1 Kaj je TrinityCore

Trinitycore je odprtokodni projekt, ki se je pod tem imenom pojavil leta 2010. Pred tem so bili še razni drugi projekti, ki so imeli podobno vizijo, vendar ni bilo prave organizacije na strani vodstva projekta, zato so se številni projekti združili skupaj in postali TrinityCore.

V začetnih fazah so bili nadgradnja ogrodju MaNGOS, ki je bil eden izmed glavnih projektov, ki so se ukvarjali z emulacijo igre World of Warcraft pred letom 2010.

Na začetku svojega obstoja je TrinityCore temeljil na objavi popravkov s strani prostovoljcev ter uporabnikov, zadnja leta pa se je preoblikoval v pravi samostojni projekt, ki ima svojo lastno skupino razvijalcev; večina izmed njih so ljudje, ki so že prej sodelovali na podobnih projektih pred pojavom TrinityCore. Dandanes je postal glavni projekt na področju emulacije igre World of Warcraft in je tudi eden izmed največjih emulacijskih projektov izmed drugih iger na splošno.

TrinityCore je celovito ogrodje, napisano v programskem jeziku C++, ter uporablja podatkovno bazo MySQL za shranjevanje vseh občutljivih, pogosto spreminjajočih se podatkov. Uporablja še lokalne podatkovne baze, shranjene v formatih DBC (v starejših različicah) ter DB2. V njih so shranjeni podatki, ki se redko spreminjajo (načeloma le kadar so izdane nove posodobitve za igro World of Warcraft) [1]. S tem ogrodjem lahko pridobimo znanja iz vseh stališč prave igre, kot so opisana v strokovnih literarnih delih [10].

V splošnem se TrinityCore razveji v dve podveji. Prva je tako imenovana master branch, ki se vedno ukvarja z najnovejšo različico igre World of Warcraft in se pretežno ukvarja z implementacijo najsodobnejših novih funkcionalnosti. Seveda pa to pomeni, da so številne funkcionalnosti nepopolne, saj se lahko, preden se popolnoma zaključijo trenutne zahteve, pojavi naslednja različica igre, zato ostaja nedodelan. Med uporabniki se šteje kot preizkusna različica zaradi njenih nepopolnosti. Druga podveja se ukvarja z razvojem tretje razširitev (*expansion*) igre World of Warcraft, in sicer Wrath of the Lich King, ki je bila izdana leta 2008. Ta podveja je mnogo bolj stabilna in dopolnjena kot master branch, vendar pa je seveda močno zastarela, saj zaostaja več kot 10 let za master podvejo, kar pomeni, da ji primanjkuje ogromno novejših funkcionalnosti, ki so bile dodane igri World of Warcraft. Vendar je ta podveja kljub vsemu bolj popularna med uporabniki, saj ima razvitih veliko več uporabniško narejenih predelav in je na splošno bolj prijazna za začetnike, saj je sistem manj zahteven.

TrinityCore omogoča, da si lahko vsakdo ogleda izvorno kodo emulatorja (*emulator*) in preveri, kako deluje katerakoli komponenta ene izmed največjih MMORPG iger na svetu, ki ima več milijonov aktivnih uporabnikov, ter lahko to kodo tudi spremeni, nadgradi ipd. Gre se torej za zelo poučni projekt, še posebej za bodoče programerje, ki bi želeli delati v industriji računalniških iger.

1.2 Kaj je emulator/emulacija

V računalništvu na splošno je emulator določena strojna ali programska oprema, ki omogoča nekemu računalniškemu sistemu, da se obnaša tako kot neki drugi sistem [2]. V praksi to po navadi pomeni, da lahko neki sistem poganja programsko opremo, ki zanj ni predvidena. Primer tega so emulatorji, ki omogočajo, da lahko na osebnih računalnikih poganjamo računalniške igrice, ki so bile izdelane za razne konzole.

V primeru TrinityCore pa emulacija pomeni, da izdelamo programsko opremo, ki zna pravilno in (skoraj) brez napak komunicirati s klientom igre World of Warcraft tako, kot to dela uradni strežnik, ki ga uporablja podjetje Blizzard Entertainment. Idealno si ekipa želi, da bi uspešno implementirali in tudi izpopolnili vse funkcionalnosti, ki jih nudi igra World of Warcraft.

1.3 Predstavitev igre World of Warcraft

World of Warcraft spada pod žanrske MMORPG-igre in je po vsej verjetnosti najbolj poznana MMORPG-igra na svetu. Izid igre je bila napovedan za leto 2001, a je izšla leta 2004 kot četrta igra v univerzi Warcraft. Zgodba se začne približno 4 leta po zadnji igri Warcraft 3 [3].

Igra je zelo znana po tem, da ima največ igralcev izmed vseh iger iz žanra MMORPG, in je obdržala ta naziv od 2004 pa vse do danes [9]. World of Warcraft je popolnoma spremenil svoj žanr, saj so mnoga podjetja poskušala ponoviti njen uspeh na veliko načinov, vključno s podvajanjem raznih lastnosti in funkcionalnosti z namenom, da bi postala naslednja uspešnica. Zanimivo je izpostaviti, da, kljub temu da igra uporablja zelo zastarelo tehnologijo ter starodobno grafiko za današnje standarde, še vedno uspeva ostati na samem vrhu v številu igralcev.

Višek igralcev so dosegli oktobra leta 2010, ko so imeli 12 milijonov dnevno aktivnih igralcev. V razširitvi Warlords of Draenor, ki je bila zdaleč najmanj priljubljena, je igra množično izgubila igralce, in je padla na 5,5 milijona igralcev, vendar so z izrednim uspehom naslednje razširitve

Legion uspeli privabiti igralce nazaj. World of Warcraft je prav tako v Guinnessovi knjigi rekordov kot najbolj priljubljena MMORPG igra po aktivnih naročnikih.

| Ekspanzija | Leto izdaje |
|------------------------|-----------------------------|
| World of Warcraft | November 2004 |
| The Burning Crusade | Januar 2007 |
| Wrath of the Lich King | November 2008 |
| Cataclysm | December 2010 |
| Mists of Pandaria | September 2012 |
| Warlords of Draenor | November 2014 |
| Legion | Avgust 2016 |
| Battle for Azeroth | September 2018 (predvidoma) |

Slika 1.1: Razširitve World of Warcraft.

Kljub vsem izdanim razširitvam (slika 1.1) pa je podjetje Blizzard Entertainment napovedalo razvoj klasičnih strežnikov, ki bodo ponujali možnost igranja originalnega World of Warcrafta iz leta 2004. To se je zgodilo zaradi množičnih peticij, ki so presegle 100.000 podpisov, da igralci želijo znova doživeti izkušnjo originalne igre, saj jim niso všeč mnoge spremembe, ki so bile narejene skozi čas.

1.3.1 Incident »Corrupted Blood«

Za konec predstavitev igre World of Warcraft si bomo ogledali še eno izmed zanimivosti, ki se je zgodila leta 2005.

Zaradi napake na zadnjem nasprotniku Hakkarju, ki je uporabljal določeno sposobnost (*spell*), imenovano »Corrupted Blood«, je bila v bistvu podobna nalezljivi bolezni in se je širila na druge igralce blizu tistega, ki je bil okužen. To stanje naj bi sicer trajalo le nekaj sekund in le v določenem območju, vendar je bilo možno zaradi nepredvidene napake priti v glavna mesta igre in okužiti celotno prebivalstvo. Napaka je sicer trajala le en teden, vendar je ta pojav privabilo pozornost epidemiologov, ki so videli potencial za simulacijo širjenja prave epidemije s pomočjo

uporabe MMORPG iger. Leta 2007 je epidemiolog Ran D Balicer izdal članek, v katerem je primerjal podobnosti te virtualne epidemije z nekaterimi pravimi epidemijami tistega časa. Zatrdil je, da bi lahko postale masivne računalniške igre napredna platforma za modeliranje nalezljivih bolezni. Inštitucija Centers for Disease Control and Prevention je kontaktirala podjetje Blizzard Entertainment in jih pozvala, naj posreduje statistične podatke o epidemiji Corrupted Blood, vendar so dobili le odgovor o tem, da je šlo za nepredvideno napako.

Incident Corrupted Blood je bil opisan kot ponesrečen, vendar čudovit za preučevanje izvora bolezni. Primerjan je bil z epidemijo v resničnem življenju, saj je izvirala v oddaljenem in izoliranem kraju in je bila prenesena od naključnih obiskovalcev v velike regije z veliko prebivalcev. Nositelji so bili ljudje (v vlogi igralcev) ter živali (v vlogi objektov tipa NPC), prenosljiva je bila preko bližanja z drugimi živimi bitji. Seveda pa so bile tudi bistvene razlike od resničnega življenja, saj so igralci lahko videli, kdo je okužen in so se lahko osebi izognili, kar v resničnosti ni možno [4].

1.4 Zakaj je TrinityCore izbral World of Warcraft

Na trgu obstaja veliko novejših iger z veliko bolj sodobno grafiko in novejšo tehnologijo kot World of Warcraft, zato se lahko vprašamo, zakaj je TrinityCore izbral prav to igro.

Razlogov je precej, čeprav igra ni sodobna:

- Priljubljenost igre in število aktivnih naročnikov. World of Warcraft je še vedno najbolj igrana in priljubljena MMORPG igra na trgu, kar pomeni, da je veliko več povpraševanja po emulatorjih za to igro kot druge manj priljubljene.
- Preprostost. Za razvijalce emulatorjev je starejša tehnologija v resnici prednost, saj je veliko lažje izdelati emulator za manj obsežno in manj zahtevno tehnologijo, kot pa nekaj zelo novega, bolj obsežnega in seveda z manj dostopne dokumentacije.
- Obsežnost igre. Kljub temu, da je World of Warcraft stara igra, ima izredno velik nabor vseh vrst objektov, kar je zelo privlačna lastnost za uporabnike, ki želijo kreirati nove objekte s pomočjo obstoječih gradnikov, ki so na voljo v igri. Prav tako je svet zelo obsežen in omogoča precej preprosto spreminjanje le-tega.
- Preprosta implementacija sprememb. Podatki in datoteke so organizirani na način, ki omogoča dokaj preprost način za spreminjanje obstoječih ali dodajanje novih podatkov,

saj so praktično vsi gradniki igre shranjeni v strežniški bazi MySQL ali pa lokalnih bazah DBC in DB2. To pomeni, da je možno na zelo preprost način spremeniti skoraj karkoli znotraj igre, brez programiranja ali drugih zahtevnih tehnik.

- Veliko drugih iger, še posebej novejše igre, kriptira večino pomembnih datotek, povezanih z igro, kar močno uteži ali nasploh onemogoči učinkovito emuliranje.

1.5 Moja vloga pri projektu TrinityCore

Pri projektu TrinityCore sodelujem pretežno v vlogi razvijalca za podatkovne baze ter, kot tako imenovani upravitelj igre (*game master*) Občasno sodelujem tudi pri izdelovanju novih funkcionalnosti neposredno v jedru strežniške programske opreme TrinityCore, vendar gre tukaj po navadi za prilagoditve obstoječe kode ali pa izdelovanje manjših skript, ki implementirajo dodatne funkcionalnosti, ki jih določen uporabnik želi imeti na svoji instanci (*instance*) TrinityCore.

Kot razvijalec podatkovnih baz se pretežno ukvarjam z dodajanjem vseh vrst objektov v strežnik TrinityCore; to so torej NPC, GOB, predmeti (*items*), dogodivščine (*quests*) itd. Seveda pa je v podatkovni bazi mogoče več kot le dodajanje in spreminjanje objektov, temveč lahko tudi konfiguriramo množico nastavitvev, kot so začetne pozicije za vsakega novega igralca, pooblaščen ukazi na voljo vsakemu igralcu, začetni predmeti za vsako možno kombinacijo rase (*race*) in razreda (*class*) in še veliko drugih opravil.

Kot upravitelj igre pa se ukvarjam kot moderator znotraj same instance igre, kar pomeni odgovarjanje na vprašanja, nudenje pomoči v primeru težav ali napak, organiziranje dogodkov za druženje znotraj igre, sodelovanje pri odločanju o novih funkcionalnostih za implementacijo in splošna skrb za stabilnost sistema in njegovo brezhibno delovanje. Praktične primere mojega lastnega dela pri projektu TrinityCore si bomo ogledali v 3. poglavju.

1.6 Cilj diplomske naloge

Cilj diplomske naloge je širjenje vednosti o obstoju projekta TrinityCore z namenom, da bi ga v bodoče pričelo uporabljati več ljudi, ki se strokovno ukvarjajo na področju razvoja iger ali v podobnih vlogah, katere se uporabljajo v namen razvoja in upravljanja projekta, kot so programerji, administratorji podatkovnih baz, sistemski administratorji, ki lahko nastopajo v vlogi upravitelja igre.

Pri projektu TrinityCore sodelujem že več kot 6 let, v tem času sem kreiral že več tisoč novih objektov v podatkovni bazi MySQL ter v lokalnih bazah DBC in DB2. Prav tako sem izdelal do sedaj že več kot 50 novih skript v jedru projekta, od tega tudi 10 večjih skript, ki implementirajo bolj zahtevne funkcionalnosti, oz. le te v večjem deležu spremenijo. Te skripte so v večini namenjene za strežnike RP, ki imajo zelo drugačna pravila kot splošni strežniki igre World of Warcraft. Poleg tega sem v okviru diplomske naloge ustvaril tudi močno namensko spletno stran, ki je namenjena za upravljanje z igralci strežnikov RP, ki je napisana v spletni tehnologiji ASP.net.

Nazadnje pa pri projektu aktivno sodelujem še kot upravitelj igre na strežnikih uporabnikov projekta, kjer sodelujem kot celovita podpora igralcev znotraj same igre (odgovarjanje na vprašanja, pomoč ob napakah in težavah, nudenje pomoči, kadar igralci ne vedo kako dokončati določeno dogodivščino ali druge vsebine znotraj igre).

Praktični primeri implementacije novih objektov ter podrobna predstavitev mojega dela pri projektu TrinityCore je predstavljeno v 3. poglavju diplomske naloge.

2 Arhitektura TrinityCore

2.1 DBC in DB2 za hranjenje statičnih podatkov

Poleg podatkovne baze za hranjenje podatkov, ki se med izvajanjem pogostokrat spreminjajo, TrinityCore uporablja še podatkovne baze hranjene v DBC (podveja Wrath of the Lich King) in DB2 (podveja master od razširitve Warlords of Draenor dalje).

Tako DBC kot DB2 se uporabljata za shranjevanje podatkov, ki se le redko spreminjajo, in sicer le takrat, ko so izdane nove posodobitve za igro World of Warcraft, kar je po navadi enkrat na teden, razen v izrednih primerih.

2.1.1 Podatkovna baza DBC

Format DBC je bil razvit pri podjetju Microsoft. Ponuja enake funkcionalnosti kot druge podatkovne baze, in sicer organizacija tabel, relacij in pogledov. Glavna slabost uporaba teh datotek je distribucija posodobitev, saj moramo vsako spremembo spraviti k vsakemu odjemalcu, saj so datoteke shranjene tako na strežniku od World of Warcraft kot tudi pri vsakem igralcu [5]. V razširitvi Wrath of the Lich King je imela igra 247 DBC datotek za shrambo lokalnih podatkov.

2.1.2 Podatkovna baza DB2

DB2 je novejši model podatkovne baze razvit pri podjetju IBM [6]. Čeprav nudi številčne prednosti pred DBC pa za namen, ki ga uporablja igra World of Warcraft, dejansko zaenkrat niti ni bil potreben. Morda bodo katere od funkcionalnosti, ki jih ponuja, postale potrebne v prihodnosti igra, vendar zaenkrat bi bilo načeloma vseeno, če bi še vedno uporabljali format DBC.

V razširitvi Legion ima igra že kar 606 DB2 podatkovnih datotek za shrambo lokalnih podatkov, kar je skoraj trikrat več kot v razširitvi Wrath of the Lich King.

- DB/Achievement
- DB/Achievement_Category
- DB/Achievement_Criteria
- DB/AnimationData
- DB/AreaGroup
- DB/AreaPOI
- DB/AreaTable
- DB/AreaTrigger
- DB/AttackAnimKits
- DB/AttackAnimTypes
- DB/AuctionHouse
- DB/BankBagSlotPrices
- DB/BannedAddOns
- DB/BarberShopStyle
- DB/BattlemasterList
- DB/CameraShakes
- DB/Cfg_Categories
- DB/Cfg_Configs
- DB/CharacterFacialHairStyles
- DB/CharBaseInfo
- DB/CharHairGeosets
- DB/CharHairTextures
- DB/CharSections
- DB/CharStartOutfit
- DB/CharTitles
- DB/CharVariations
- DB/ChatChannels
- DB/ChatProfanity
- DB/ChrClasses
- DB/ChrRaces
- DB/CinematicCamera
- DB/CinematicSequences
- DB/CreatureDisplayInfo
- DB/CreatureDisplayInfoExtra
- DB/CreatureFamily
- DB/CreatureModelData
- DB/CreatureMovementInfo
- DB/CreatureSoundData
- DB/CreatureSpellData
- DB/CreatureType
- DB/CurrencyCategory
- DB/CurrencyTypes
- DB/DanceMoves
- DB/DeathThudLookups
- DB/DeclinedWord
- DB/DeclinedWordCases
- DB/DestructibleModelData
- DB/DungeonEncounter
- DB/DungeonMap
- DB/DungeonMapChunk
- DB/DurabilityCosts
- DB/DurabilityQuality
- DB/Emotes
- DB/EmotesText
- DB/EmotesTextData
- DB/EmotesTextSound
- DB/EnvironmentalDamage
- DB/Exhaustion
- DB/Faction
- DB/FactionGroup
- DB/FactionTemplate
- DB/FileData
- DB/FootprintTextures
- DB/FootstepTerrainLookup
- DB/GameObjectArtKit
- DB/GameObjectDisplayInfo
- DB/SpellVisualKitAreaModel
- DB/SpellVisualKitModelAttach
- DB/SpellVisualPrecastTransitions
- DB/GameTables
- DB/GameTips
- DB/GemProperties
- DB/GlyphProperties
- DB/GlyphSlot
- DB/GMSurveyAnswers
- DB/GMSurveyCurrentSurvey
- DB/GMSurveyQuestions
- DB/GMSurveySurveys
- DB/GMTicketCategory
- DB/GroundEffectDoodad
- DB/GroundEffectTexture
- DB/GtBarberShopCostBase
- DB/GtChanceToMeleeCrit
- DB/GtChanceToMeleeCritBase
- DB/GtChanceToSpellCrit
- DB/GtChanceToSpellCritBase
- DB/GtCombatRatings
- DB/GtNPCManaCostScaler
- DB/GtOCTClassCombatRatingScalar
- DB/GtOCTRegenHP
- DB/GtOCTRegenMP
- DB/GtRegenHPPerSpt
- DB/GtRegenMPPerSpt
- DB/HelmetGeosetVisData
- DB/HolidayDescriptions
- DB/HolidayNames
- DB/Holidays
- DB/Item
- DB/ItemBagFamily
- DB/ItemClass
- DB/ItemCondExtCosts
- DB/ItemDisplayInfo
- DB/ItemExtendedCost
- DB/ItemGroupSounds
- DB/ItemPetFood
- DB/ItemPurchaseGroup
- DB/ItemRandomProperties
- DB/ItemRandomSuffix
- DB/ItemSet
- DB/ItemSubClass
- DB/ItemSubClassMask
- DB/ItemVisualEffects
- DB/ItemVisuals
- DB/Languages
- DB/LanguageWords
- DB/LfgDungeonGroup
- DB/LfgDungeons
- DB/Light
- DB/LightFloatBand
- DB/LightIntBand
- DB/LightParams
- DB/LightSkybox
- DB/LiquidMaterial
- DB/LiquidType
- DB>LoadingScreens
- DB>LoadingScreenTaxiSplines
- DB/Lock
- DB/LockType
- DB/MailTemplate
- DB/Map
- DB/MapDifficulty
- DB/Material
- DB/Movie
- DB/MovieFileData
- DB/MovieVariation
- DB/NameGen
- DB/TerrainType
- DB/TerrainTypeSounds
- DB/TotemCategory
- DB/TransportAnimation
- DB/TransportPhysics
- DB/NamesProfanity
- DB/NamesReserved
- DB/NPCSounds
- DB/ObjectEffect
- DB/ObjectEffectGroup
- DB/ObjectEffectModifier
- DB/ObjectEffectPackage
- DB/ObjectEffectPackageElem
- DB/OverrideSpellData
- DB/Package
- DB/PageTextMaterial
- DB/PaperDollItemFrame
- DB/ParticleColor
- DB/PetitionType
- DB/PetPersonality
- DB/PowerDisplay
- DB/PvpDifficulty
- DB/QuestFactionReward
- DB/QuestInfo
- DB/QuestSort
- DB/QuestXP
- DB/RandPropPoints
- DB/Resistances
- DB/ScalingStatDistribution
- DB/ScalingStatValues
- DB/ScreenEffect
- DB/ServerMessages
- DB/SheatheSoundLookups
- DB/SkillCostsData
- DB/SkillLine
- DB/SkillLineAbility
- DB/SkillLineCategory
- DB/SkillRaceClassInfo
- DB/SkillTiers
- DB/SoundAmbience
- DB/SoundEmitters
- DB/SoundEntries
- DB/SoundFilter
- DB/SoundFilterElem
- DB/SoundProviderPreferences
- DB/SoundSamplePreferences
- DB/SoundWaterType
- DB/SpamMessages
- DB/Spell
- DB/SpellCastTimes
- DB/SpellCategory
- DB/SpellChainEffects
- DB/SpellDescriptionVariables
- DB/SpellDifficulty
- DB/SpellDispelType
- DB/SpellDuration
- DB/SpellEffectCameraShakes
- DB/SpellEffectNames
- DB/SpellFocusObject
- DB/SpellIcon
- DB/SpellItemEnchantment
- DB/SpellItemEnchantmentCondition
- DB/SpellMechanic
- DB/SpellMissile
- DB/SpellMissileMotion
- DB/SpellRadius
- DB/SpellRange
- DB/SpellRuneCost
- DB/SpellShapeshiftForm
- DB/SpellVisual
- DB/SpellVisualEffectName
- DB/SpellVisualKit
- DB/WeaponImpactSounds
- DB/WeaponSwingSounds2
- DB/Weather
- DB/WMOAreaTable
- DB/WorldChunkSounds

Slika 2.1: Datoteke DBC v razširitvi Wrath of the Lich King.

2.1.3 Pregled najbolj pomembnih DBC/DB2 datotek

Kot lahko vidimo (slika 2.1), igra World of Warcraft vsebuje precejšnje število teh lokalnih podatkovnih zbirk. V nadaljevanju si bomo ogledali funkcionalnost le nekaterih najpomembnejših oz. tistih, ki so najbolj zanimive za razvijalce.

- CharSections: vsebuje podatke o izbiri, kako izgledajo igralniški liki (*player characters*) in NPC-ji za vsako možno raso v igri World of Warcraft. Vsebuje podatke o očeh, laseh, obrazih, barvi kože, o bradi in nakitu.
- CreatureModelData: vsebuje podatke o vseh modelih za vsak NPC, ki obstaja v svetu. Tukaj določimo lastnosti, kot so kolizija, velikost, kakšne sledi pušča za seboj, ko se premika in še posebne zastavice, ki določijo obnašanje modela, na primer, če je model vozilo, ima kolesa ipd.
- Faction: vsebuje podatke o vseh frakcijah (*fraction*) in njihovi hierarhiji. Na primer frakcija »Undercity« je podrejena frakciji »Horde«. Tukaj prav tako določimo vse odnose med posameznimi frakcijami, na primer »Horde« in »Aliance« sta si nasprotni, kar pomeni, da če se srečata dva NPC-ja iz nasprotnih frakcij, se bosta napadla.
- Item: vsebuje podatke o vseh predmetih, ki jih lahko pridobimo v svetu. To vsebuje orožje, oklepe, napitke, knjige, pisma in še mnogo drugih. V razširitvi Legion igra vsebuje 127666 predmetov.
- Map: Vsebuje podatke o vseh mapah po svetu. Za vsako mapo se določijo številni podatki, in sicer: vrsta mape (instanca, normalna mapa, PVP-mapa itd.), ime mape, kateri frakciji pripada mapa (načeloma Horde, Aliance ali pa nevtralna), zahteve za vstop – po navadi mora biti igralniški lik določene stopnje (level), x- in y-koordinati mape, nastavitve dnevnega cikla za mapo, kateri razširitvi pripada mapa itd.
- Tabela Spell in vse druge tabele s predpono Spell skupaj določajo vse lastnosti o vseh sposobnostih, torej njihova imena, opisi, kdaj se sprožijo, kaj naredijo, ko se sprožijo, kakšne so zahteve, da se lahko sprožijo, vizualni učinki, ko se sprožijo, in še veliko drugih lastnosti. V razširitvi Legion igra vsebuje 178.613 sposobnosti.

| ID | FileDataID | Class | SubClass | SoundOverride | Material | InventoryType | Sheath | GroupSoundsID |
|----|------------|-------|----------|---------------|----------|---------------|--------|---------------|
| 17 | 132759 | 4 | 4 | 255 | 6 | 4 | 3 | 11 |
| 25 | 135274 | 2 | 7 | 255 | 1 | 21 | 3 | 8 |
| 35 | 135145 | 2 | 10 | 255 | 2 | 17 | 2 | 13 |
| 36 | 133478 | 2 | 4 | 255 | 2 | 21 | 3 | 8 |
| 37 | 132402 | 2 | 0 | 255 | 1 | 21 | 3 | 8 |
| 38 | 135009 | 4 | 0 | 255 | 7 | 4 | 0 | 7 |
| 39 | 134582 | 4 | 4 | 255 | 6 | 7 | 0 | 8 |
| 40 | 132540 | 4 | 4 | 255 | 6 | 8 | 0 | 8 |
| 41 | 132494 | 4 | 0 | 255 | 7 | 6 | 0 | 7 |
| 42 | 132494 | 4 | 0 | 255 | 7 | 6 | 0 | 7 |
| 43 | 132535 | 4 | 4 | 255 | 6 | 8 | 0 | 8 |
| 44 | 134582 | 4 | 4 | 255 | 6 | 7 | 0 | 8 |
| 45 | 135005 | 4 | 0 | 255 | 7 | 4 | 0 | 7 |
| 46 | 132495 | 4 | 0 | 255 | 7 | 6 | 0 | 7 |
| 47 | 132539 | 4 | 2 | 255 | 7 | 8 | 0 | 7 |
| 48 | 134582 | 4 | 2 | 255 | 7 | 7 | 0 | 7 |
| 49 | 135009 | 4 | 0 | 255 | 7 | 4 | 0 | 7 |
| 50 | 132493 | 4 | 0 | 255 | 7 | 6 | 0 | 7 |
| 51 | 132538 | 4 | 1 | 255 | 7 | 8 | 0 | 7 |
| 52 | 134582 | 4 | 1 | 255 | 7 | 7 | 0 | 7 |
| 53 | 135005 | 4 | 0 | 255 | 7 | 4 | 0 | 7 |
| 54 | 132493 | 4 | 0 | 255 | 7 | 6 | 0 | 7 |
| 55 | 132543 | 4 | 1 | 255 | 7 | 8 | 0 | 7 |
| 56 | 132664 | 4 | 1 | 255 | 7 | 20 | 0 | 7 |
| 57 | 132665 | 4 | 1 | 255 | 7 | 20 | 0 | 7 |
| 58 | 132492 | 4 | 0 | 255 | 7 | 6 | 0 | 7 |
| 59 | 132535 | 4 | 1 | 255 | 7 | 8 | 0 | 7 |
| 60 | 132724 | 4 | 2 | 255 | 8 | 5 | 0 | 7 |
| 61 | 443401 | 4 | 2 | 255 | 8 | 7 | 0 | 7 |
| 77 | 135006 | 4 | 1 | 255 | 7 | 5 | 0 | 7 |
| 79 | 443400 | 4 | 1 | 255 | 7 | 7 | 0 | 7 |
| 80 | 132543 | 4 | 1 | 255 | 7 | 8 | 0 | 7 |
| 85 | 135011 | 4 | 2 | 255 | 8 | 5 | 0 | 7 |
| 86 | 132493 | 4 | 0 | 255 | 7 | 6 | 0 | 7 |
| 87 | 132535 | 4 | 0 | 255 | 7 | 8 | 0 | 7 |
| 88 | 134582 | 4 | 1 | 255 | 7 | 7 | 0 | 7 |
| 89 | 135011 | 4 | 0 | 255 | 7 | 4 | 0 | 7 |
| 90 | 132494 | 4 | 0 | 255 | 7 | 6 | 0 | 7 |
| 91 | 134582 | 4 | 1 | 255 | 7 | 7 | 0 | 7 |
| 92 | 132535 | 4 | 0 | 255 | 7 | 8 | 0 | 7 |
| 93 | 135005 | 4 | 0 | 255 | 7 | 4 | 0 | 7 |
| 94 | 132494 | 4 | 0 | 255 | 7 | 6 | 0 | 7 |

Files

Item.db2 - Legion 7.3.0 (24793)

Filter

Filter

Build

Statistics

Current File:

Definition:

Current Cell:

Stats:

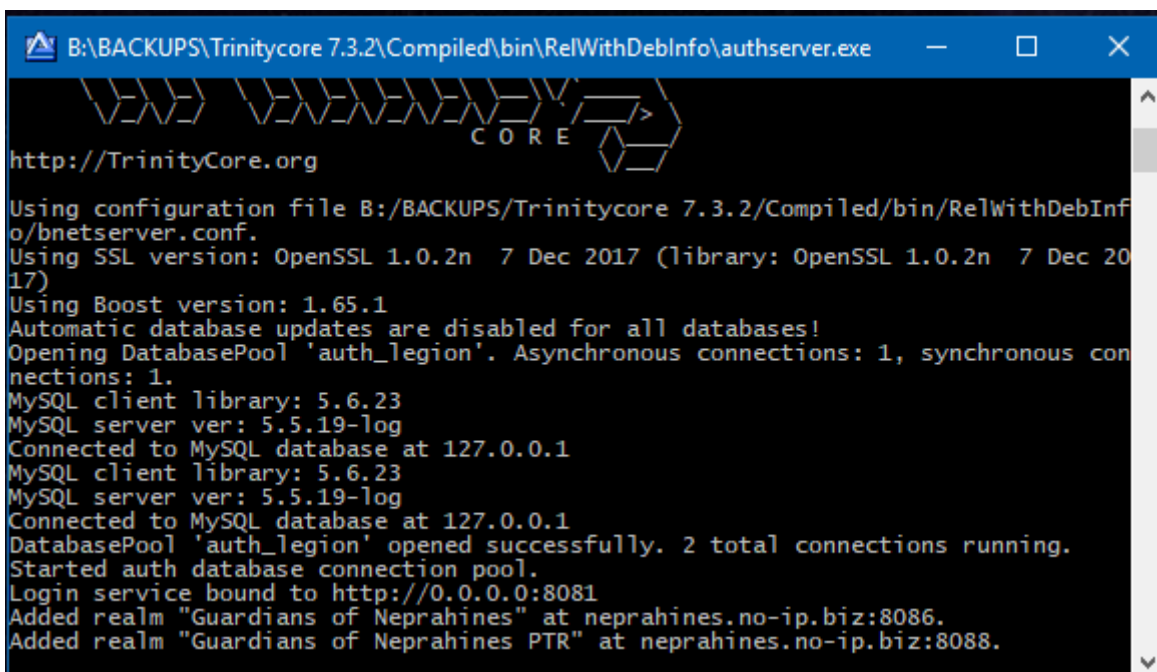
Slika 2.2: Primer datoteke DB2.

Kot lahko pričakujemo, igre dandanes vsebujejo veliko količino podatkov (slika 2.2). Na sliki lahko vidimo, kako izgledajo podatkovne baze DB2 in kako so organizirane na primeru ene izmed največjih DB2-datotek.

2.2 Strežnika Auth/Bnet in World

Strežnika Auth in World sta jedro TrinityCore, saj sta dejanska emulatorja strežnika od igre World of Warcraft. V novejših različicah World of Warcraft se je strežnik Auth preimenoval v strežnik Bnet. Njegove naloge še vedno ostajajo enake, zato bomo v tem poglavju uporabljali izključno ime strežnik Auth. Oba strežnika sta napisana v programskem jeziku C++. V tem poglavju si bomo na kratko ogledali njihovo delovanje in njihove glavne naloge.

2.2.1 Strežnik Auth

The image shows a Windows command prompt window titled "B:\BACKUPS\Trinitycore 7.3.2\Compiled\bin\RelWithDebInfo\authserver.exe". The console output displays the TrinityCore logo and the following text:

```
http://TrinityCore.org
Using configuration file B:/BACKUPS/Trinitycore 7.3.2/Compiled/bin/RelWithDebInfo/bnetserver.conf.
Using SSL version: OpenSSL 1.0.2n 7 Dec 2017 (library: OpenSSL 1.0.2n 7 Dec 2017)
Using Boost version: 1.65.1
Automatic database updates are disabled for all databases!
Opening DatabasePool 'auth_legion'. Asynchronous connections: 1, synchronous connections: 1.
MySQL client library: 5.6.23
MySQL server ver: 5.5.19-log
Connected to MySQL database at 127.0.0.1
MySQL client library: 5.6.23
MySQL server ver: 5.5.19-log
Connected to MySQL database at 127.0.0.1
DatabasePool 'auth_legion' opened successfully. 2 total connections running.
Started auth database connection pool.
Login service bound to http://0.0.0.0:8081
Added realm "Guardians of Neprahines" at neprahines.no-ip.biz:8086.
Added realm "Guardians of Neprahines PTR" at neprahines.no-ip.biz:8088.
```

Slika 2.3: Izgled strežnika Auth ko ga odpremo.

Strežnik Auth je manj zahteven od strežnika World, saj ima precej manj nalog kot slednji. Njegova glavna naloga je vzpostavitev in vzdrževanje komunikacije, tako da se igralci oz. klienti dejansko lahko sploh prijavijo v sistem. Ko pride nova zahteva za prijavo, strežnik preveri klientove podatke (uporabniško ime in geslo) in če je le to uspešno, ga preusmeri na strežnik World. Kot lahko vidimo na zgornji sliki (slika 2.3), imamo v našem primeru dva kanala (*realm*) na strežnik World, kar dejansko pomeni, da na tem računalniku tečeta dva strežnika World. To je zelo pogosta praksa; pravi strežnik World of Warcraft ima več kot 10 kanalov, ki so ves čas

aktivni. V našem primeru imamo le glavni kanal »Guardians of Neprahines« in kanal za razvijalce in preizkušanje »Guardians of Neprahines PTR«, ki je dostopen izključno razvijalcem in uporabnikom s pooblaščenimi pravicami (upravitelji igre, administratorji). Za svoje delovanje strežnik Auth uporablja izključno podatkovno bazo Auth, ostalih dveh (ali treh na novejših različicah) pa nikoli ne potrebuje za delovanje.

```

1 #####
2 # Trinity Core Auth Server configuration file #
3 #####
4 [bnetsserver]
5
6 #####
7 # SECTION INDEX
8 #
9 #   EXAMPLE CONFIG
10 #   AUTH SERVER SETTINGS
11 #   MYSQL SETTINGS
12 #   UPDATE SETTINGS
13 #   LOGGING SYSTEM SETTINGS
14 #
15 #####
16
17 #####
18 # EXAMPLE CONFIG
19 #
20 #   Variable
21 #       Description: Brief description what the variable is doing.
22 #       Important:   Annotation for important things about this variable.
23 #       Example:    "Example, i.e. if the value is a string"
24 #       Default:    10 - (Enabled|Comment|Variable name in case of grouped config options)
25 #                 0  - (Disabled|Comment|Variable name in case of grouped config options)
26 #
27 # Note to developers:
28 # - Copy this example to keep the formatting.
29 # - Line breaks should be at column 100.
30 #####
31
32 #####
33 # AUTH SERVER SETTINGS
34 #
35 #   LogsDir
36 #       Description: Logs directory setting.
37 #       Important:   LogsDir needs to be quoted, as the string might contain space characters.
38 #                   Logs directory must exists, or log file creation will be disabled.
39 #       Default:    "" - (Log files will be stored in the current path)
40
41 LogsDir = ""
42
43 #
44 #   MaxPingTime
45 #       Description: Time (in minutes) between database pings.
46 #       Default:    30
47
48 MaxPingTime = 30
49
50 #
51 #   BattlenetPort
52 #       Description: TCP port to reach the auth server for battle.net connections.
53 #       Default:    1119
54

```

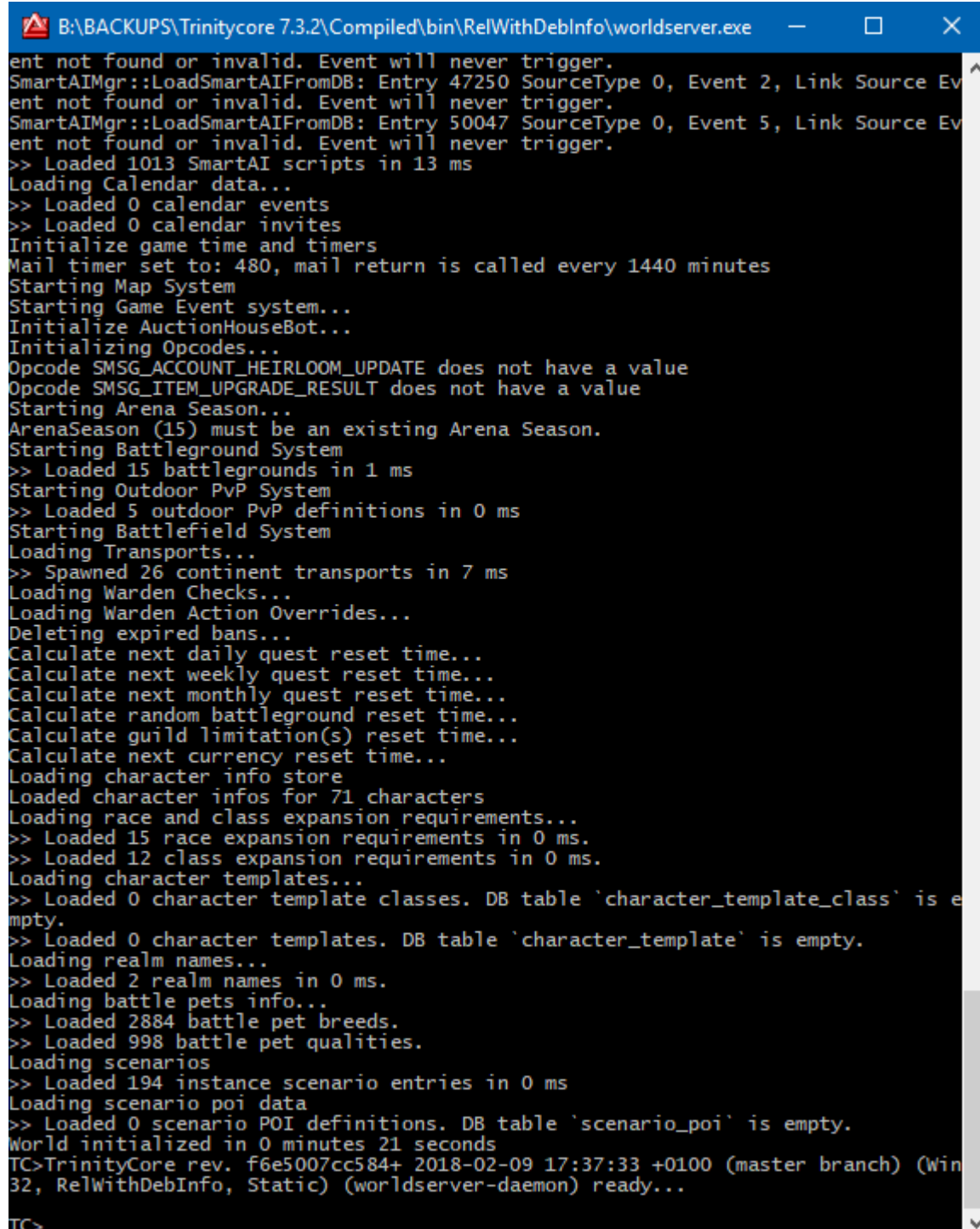
Slika 2.4: Konfiguracijska datoteka za strežnik Auth.

Za svoje delovanje potrebuje strežnik Auth tudi konfiguracijsko datoteko (slika 2.4), v kateri lahko spremenimo skoraj vse funkcionalnosti, ki jih le-ta ponuja. Če konfiguracijska datoteka ne obstaja ali pa če je le narobe zasnovana (na primer, če manjka kakšna spremenljivka), se bo strežnik Auth sesul in bo zabeležil napako, dokler je ne odpravimo. Na srečo pa so napake zastavljene zelo dobro, tako da zna strežnik točno najti, katera spremenljivka manjka, in to tudi zabeležiti.

Oglejmo si nekaj najpomembnejših lastnosti, ki jih lahko spremenimo v konfiguracijski datoteki:

- kam naj se beležijo in shranjujejo vsa poročila o delovanju strežnika (opozorila, napake);
- kolikokrat lahko vnesemo napačno uporabniško ime in geslo, preden nas bo strežnik Auth začasno blokiral. Lahko definiramo tudi, za koliko časa nas bo blokiral, če bo blokiral le naš račun ali pa naš naslov IP, in tudi, če želimo to funkcionalnost sploh imeti vklopljeno;
- nastavitve za povezavo s podatkovno bazo Auth, ki se lahko nahaja tudi na drugem računalniku kot strežnik Auth;
- možnost samostojnih posodobitev podatkovne baze Auth, kar nam omogoča, da se vse nove spremembe na bazi od strani razvijalcev TrinityCore samostojno prenesejo in izvedejo na naši bazi;
- nastavitve naslova IP in port, na katerem bo strežnik Auth sprejemal nove povezave.

2.2.2 Strežnik World



```
B:\BACKUPS\Trinitycore 7.3.2\Compiled\bin\RelWithDebInfo\worldserver.exe
ent not found or invalid. Event will never trigger.
SmartAIMgr::LoadSmartAIFromDB: Entry 47250 SourceType 0, Event 2, Link Source Ev
ent not found or invalid. Event will never trigger.
SmartAIMgr::LoadSmartAIFromDB: Entry 50047 SourceType 0, Event 5, Link Source Ev
ent not found or invalid. Event will never trigger.
>> Loaded 1013 SmartAI scripts in 13 ms
Loading Calendar data...
>> Loaded 0 calendar events
>> Loaded 0 calendar invites
Initialize game time and timers
Mail timer set to: 480, mail return is called every 1440 minutes
Starting Map System
Starting Game Event system...
Initialize AuctionHouseBot...
Initializing Opcodes...
Opcode SMSG_ACCOUNT_HEIRLOOM_UPDATE does not have a value
Opcode SMSG_ITEM_UPGRADE_RESULT does not have a value
Starting Arena Season...
ArenaSeason (15) must be an existing Arena Season.
Starting Battleground System
>> Loaded 15 battlegrounds in 1 ms
Starting Outdoor PvP System
>> Loaded 5 outdoor PvP definitions in 0 ms
Starting Battlefield System
Loading Transports...
>> Spawned 26 continent transports in 7 ms
Loading Warden Checks...
Loading Warden Action Overrides...
Deleting expired bans...
Calculate next daily quest reset time...
Calculate next weekly quest reset time...
Calculate next monthly quest reset time...
Calculate random battleground reset time...
Calculate guild limitation(s) reset time...
Calculate next currency reset time...
Loading character info store
Loaded character infos for 71 characters
Loading race and class expansion requirements...
>> Loaded 15 race expansion requirements in 0 ms.
>> Loaded 12 class expansion requirements in 0 ms.
Loading character templates...
>> Loaded 0 character template classes. DB table `character_template_class` is e
mpty.
>> Loaded 0 character templates. DB table `character_template` is empty.
Loading realm names...
>> Loaded 2 realm names in 0 ms.
Loading battle pets info...
>> Loaded 2884 battle pet breeds.
>> Loaded 998 battle pet qualities.
Loading scenarios
>> Loaded 194 instance scenario entries in 0 ms
Loading scenario poi data
>> Loaded 0 scenario POI definitions. DB table `scenario_poi` is empty.
World initialized in 0 minutes 21 seconds
TC>TrinityCore rev. f6e5007cc584+ 2018-02-09 17:37:33 +0100 (master branch) (Win
32, RelWithDebInfo, Static) (worldserver-daemon) ready..
TC>
```

Slika 2.5: Izgled strežnika World, ko ga odpremo.

V nasprotju s strežnikom Auth pa je strežnik World veliko bolj zahteven in vsebuje veliko več funkcionalnosti, saj mora le-ta upravljati s celotnim svetom, vsemi igralci, ki izvajajo akcije, vsemi dogodki po svetu, vremenom, NPC-ji in še številne druge naloge. Strežnik World prevzame vso vlogo podpore za igralce, takoj ko se ti uspešno prijavijo v igro preko strežnika Auth. Po navadi se uporabljajo številni strežniki World sočasno, najmanj eden za igralce in še eden za razvijalce, kjer lahko testirajo nove funkcionalnosti in spreminjajo svet, brez da bi to opazili igralci, ki lahko sočasno igrajo na strežniku. Še posebej je to pomembno, kadar pride do napak, ali pa se strežnik sesuje.

Če uporabljamo vsaj 2 strežnika, bodo lahko igralci še naprej igrali tudi, če se sesuje strežnik za razvijalce.

```

115 #
116 # LoginDatabase.WorkerThreads
117 # WorldDatabase.WorkerThreads
118 # CharacterDatabase.WorkerThreads
119 # HotfixDatabase.WorkerThreads
120 # Description: The amount of worker threads spawned to handle asynchronous (delayed) MySQL
121 # statements. Each worker thread is mirrored with its own connection to the
122 # MySQL server and their own thread on the MySQL server.
123 # Default: 1 - (LoginDatabase.WorkerThreads)
124 # 1 - (WorldDatabase.WorkerThreads)
125 # 1 - (CharacterDatabase.WorkerThreads)
126 # 1 - (HotfixDatabase.WorkerThreads)
127
128 LoginDatabase.WorkerThreads = 1
129 WorldDatabase.WorkerThreads = 1
130 CharacterDatabase.WorkerThreads = 1
131 HotfixDatabase.WorkerThreads = 1
132
133 #
134 # LoginDatabase.SynchThreads
135 # WorldDatabase.SynchThreads
136 # CharacterDatabase.SynchThreads
137 # HotfixDatabase.SynchThreads
138 # Description: The amount of MySQL connections spawned to handle.
139 # Default: 1 - (LoginDatabase.SynchThreads)
140 # 1 - (WorldDatabase.SynchThreads)
141 # 2 - (CharacterDatabase.SynchThreads)
142 # 1 - (HotfixDatabase.SynchThreads)
143
144 LoginDatabase.SynchThreads = 1
145 WorldDatabase.SynchThreads = 1
146 CharacterDatabase.SynchThreads = 2
147 HotfixDatabase.SynchThreads = 1
148
149 #
150 # MaxPingTime
151 # Description: Time (in minutes) between database pings.
152 # Default: 30
153
154 MaxPingTime = 30
155
156 #
157 # WorldServerPort
158 # Description: TCP port to reach the world server.
159 # Default: 8085
160
161 WorldServerPort = 8086
162
163 #
164 # InstanceServerPort
165 # Description: TCP port to for second world connection.
166 # Default: 8086
167
168 InstanceServerPort = 8087

```

Normal text file | length: 137.161 lines: 4.017 | Ln: 1 Col: 1 Sel: 0 | 0 | Windows (CR LF) | UTF-8 | INS

Slika 2.6: Konfiguracijska datoteka za strežnik World.

Tako kot strežnik Auth tudi strežnik World za svoje delovanje potrebuje konfiguracijsko datoteko, v kateri lahko spremenimo skoraj vse funkcionalnosti, ki jih ponuja (slika 2.6). Glede na to, da je strežnik World veliko bolj funkcionalno obsežen, je tudi konfiguracijska datoteka veliko večja in vsebuje ogromno število spremenljivk, ki jih lahko nastavimo. Tako kot pri strežniku Auth pa tudi World ne more delovati brez konfiguracijske datoteke, in se bo sesul, če le ta ne obstaja ali pa je napačno konfigurirana. Seveda pa so tudi pri strežniku World na voljo zelo podrobna sporočila o napakah, ki točno definirajo, katera spremenljivka manjka ali pa ima nesprejemljivo vrednost (na primer, pozabimo piko pri naslovu IP). Če želimo poganjati več kot en strežnik World na istem računalniku, moramo fizično imeti dve kopiji strežnika World in

njihove konfiguracyjske datoteke, saj mora vsaja instanca imeti svojo kombinacijo naslova IP in port, sicer ne more delovati pravilno in se bi drugi strežnik sesul.

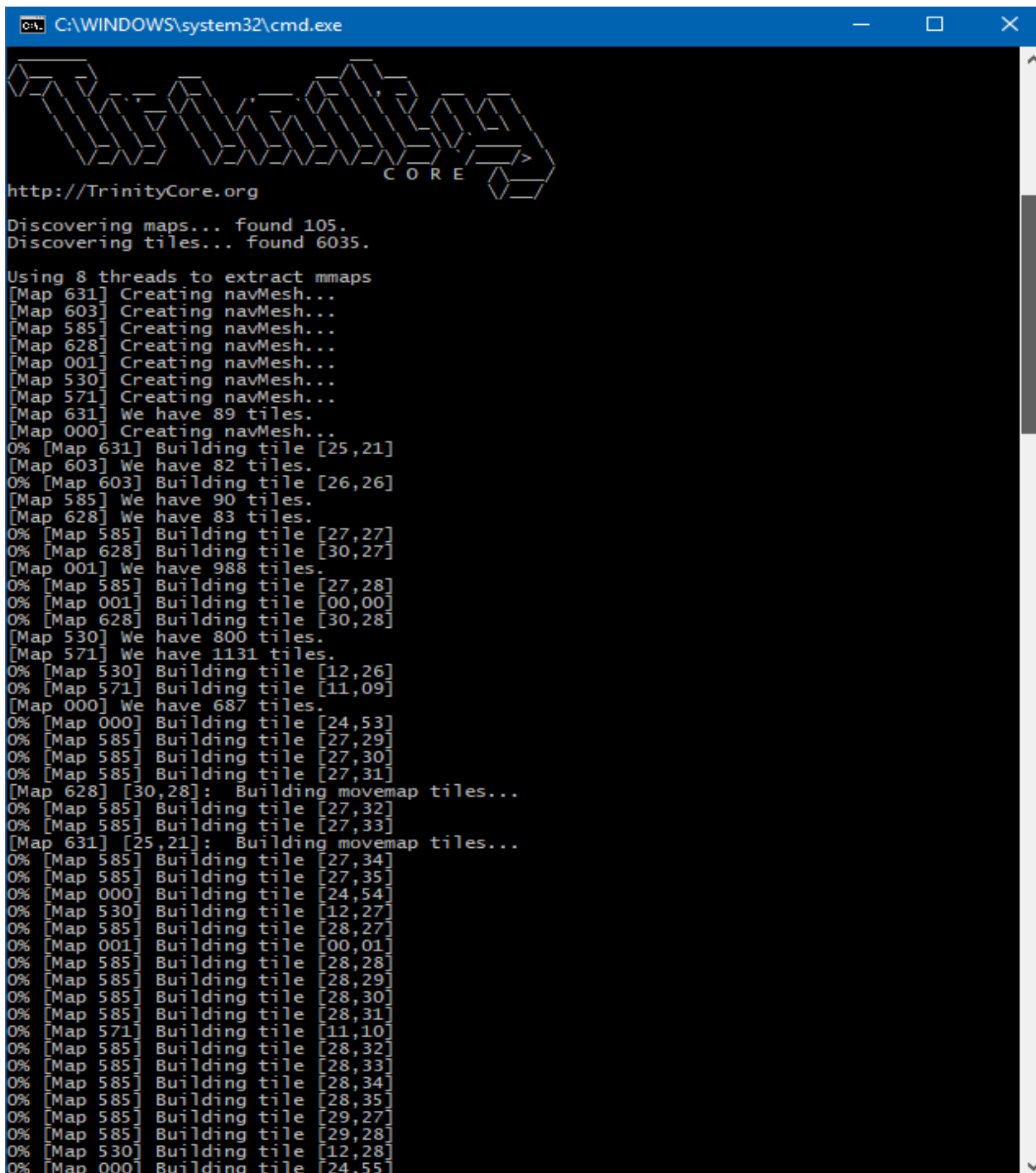
Strežnik World pri svojem delovanju uporablja vse 3 (4 pri novejših različicah) podatkovne baze. Sedaj si bomo ogledali nekaj najpomembnejših nastavitev še za strežnik World.

- Nastavitve za vse 3 (4) podatkovne baze. Te se lahko nahajajo tudi na drugem računalniku kot strežnik World.
- Lokacija, kamor naj se shranjujejo poročila o opozorilih in napakah.
- Konfiguracija naslova IP in port (mora biti unikatna za vsako instanco strežnika World, ki teče na istem računalniku).
- Koliko računalniških virov želimo alocirati strežniku World (število jeder na procesorju, število procesorjev, prioriteta procesa).
- Nastavitve za izvedbena opravila, npr. kako pogosto naj se posodablja svet, spreminja vreme, shranijo podatki v podatkovno bazo in še nekaj drugih. Vse to vpliva na to, koliko bremena bo strežnik World imel na računalnik, na katerem se izvaja. Za manj zmogljive sisteme je te nastavitve treba seveda zmanjšati.
- Veliko nastavitev o tem, če želimo določene sisteme igre World of Warcraft izklopiti, kot so recimo trgovina, arena, bojišča (*battlegrounds*) itd.
- Določene nastavitve za igralce: koliko denarja imajo, ko začnejo igrati, najmanj koliko dolgo ime morajo imeti, ali lahko ime vsebuje posebne znake, katere rase in razredi so omogočeni, koliko časa potuje pošta, preden pride do igralca, in še precej drugih.
- Možnost posodobitev za vse 3 (4) podatkovne baze, kar omogoča, da strežnik samostojno preverja, če so na voljo kakšne posodobitve za katero izmed podatkovnih baz in se te samostojno prenesejo in namestijo.
- Sistem Hotswap, ki omogoča, da lahko spreminjamo ali dodajamo skripte, brez da bi morali znova prevesti celotno jedro TrinityCore, kar močno pospeši delo razvijalcev.
- Sistem Warden za detekcijo hekerjev in goljufov. Lahko nastavimo, kako strog je sistem pri zaznavi ter kaj naj naredi, če zazna hekerja.
- Nastavitve za NPC-je, kot so: koliko izkušenj (*experience*) dajo, kako dolgo lahko sledijo igralcu, ki beži, preden se vrnejo na svojo prvotno lokacijo, konstante, s katerimi lahko na

zelo preprost način otežimo ali olajšamo celotno igro (na primer, še spremenimo konstanto življenjske točke iz 1 na 2, bodo vsi NPC-ji postali še enkrat bolj vzdržljivi).

- Razdalja, kako daleč se vidijo besedila, ki jih napišejo igralci in NPC-ji.
- Nastavitve za pooblašene igralce (npr. upravitelji igre). Lahko določimo drugačne konstante za večina lastnosti, ki smo jih določili za navadne igralce (začetna količina denarja, stopnje ipd.)
- Konfiguracija za pomoč uporabnikom (ti. ticket sistem), preko katerega lahko igralci pišejo prošnje, pripombe ali pritožbe, na katere se potem pooblašeni igralci (npr. upravitelji igre) odzovejo.
- In še veliko drugih. Konfiguracijska datoteka vsebuje okoli 500 spremenljivk, kaj jih lahko nastavimo.

2.3 Programi za ekstrahiranje podatkov iz igre



```
C:\WINDOWS\system32\cmd.exe

http://TrinityCore.org

Discovering maps... found 105.
Discovering tiles... found 6035.

Using 8 threads to extract mmap
[Map 631] Creating navMesh...
[Map 603] Creating navMesh...
[Map 585] Creating navMesh...
[Map 628] Creating navMesh...
[Map 001] Creating navMesh...
[Map 530] Creating navMesh...
[Map 571] Creating navMesh...
[Map 631] We have 89 tiles.
[Map 000] Creating navMesh...
0% [Map 631] Building tile [25,21]
[Map 603] We have 82 tiles.
0% [Map 603] Building tile [26,26]
[Map 585] We have 90 tiles.
[Map 628] We have 83 tiles.
0% [Map 585] Building tile [27,27]
0% [Map 628] Building tile [30,27]
[Map 001] We have 988 tiles.
0% [Map 585] Building tile [27,28]
0% [Map 001] Building tile [00,00]
0% [Map 628] Building tile [30,28]
[Map 530] We have 800 tiles.
[Map 571] We have 1131 tiles.
0% [Map 530] Building tile [12,26]
0% [Map 571] Building tile [11,09]
[Map 000] We have 687 tiles.
0% [Map 000] Building tile [24,53]
0% [Map 585] Building tile [27,29]
0% [Map 585] Building tile [27,30]
0% [Map 585] Building tile [27,31]
[Map 628] [30,28]: Building movemap tiles...
0% [Map 585] Building tile [27,32]
0% [Map 585] Building tile [27,33]
[Map 631] [25,21]: Building movemap tiles...
0% [Map 585] Building tile [27,34]
0% [Map 585] Building tile [27,35]
0% [Map 000] Building tile [24,54]
0% [Map 530] Building tile [12,27]
0% [Map 585] Building tile [28,27]
0% [Map 001] Building tile [00,01]
0% [Map 585] Building tile [28,28]
0% [Map 585] Building tile [28,29]
0% [Map 585] Building tile [28,30]
0% [Map 585] Building tile [28,31]
0% [Map 571] Building tile [11,10]
0% [Map 585] Building tile [28,32]
0% [Map 585] Building tile [28,33]
0% [Map 585] Building tile [28,34]
0% [Map 585] Building tile [28,35]
0% [Map 585] Building tile [29,27]
0% [Map 585] Building tile [29,28]
0% [Map 530] Building tile [12,28]
0% [Map 000] Building tile [24,55]
```

Slika 2.7: Potek ekstrakcije lokalnih podatkov iz igre.

Strežnik World za svoje delovanje potrebuje določene podatke, ki so enaki tistim, ki jih vsebuje sama igra World of Warcraft, in sicer:

- Zbirka DBC/DB2 datotek (glej poglavje 2.1). To je zbirka podatkovnih baz, ki vsebujejo podatke o vseh elementih sveta igre World of Warcraft (NPC, predmeti, igralci, vreme, itd.). Te so obvezne in strežnik World se bo sesul, če nimamo zbirke DBC/DB2.
- Mape (*maps*). Tudi mape so obvezne in se bo strežnik brez njih sesul. Mape dejansko hranijo podatke o vseh mapah po svetu in njihove fizične lastnosti, ki so potrebne, da se lahko igralci sploh premikajo po njih.
- Virtualne mape (*virtual maps - vmaps*). Virtualne mape niso obvezne in strežnik lahko tudi brez njih deluje. Virtualne mape so uporabljene za računanje razdalj, recimo kako daleč lahko igralec vidi. Praktično pa se pretežno uporabljajo za ugotavljanje, če je določene sposobnosti možno uporabiti v danem trenutku (npr. Če je med igralcem in tarčo neka ovira, recimo zid). Brez virtualnih map lahko sposobnosti uporabljamo skozi vse ovire.
- Premične mape (*movement maps – mmap*s). Tudi premične mape niso obvezne za delovanje strežnika. Uporabljajo se za dodeljevanje fizike tudi objektom, ki niso igralci (v večini primerov gre za NPC-je), saj klient sam ne obravnava kolizije NPC-objektov. Prav tako izboljšajo tudi generiranje poti iz točke A v točko B (recimo, če se določen NPC mora nekam premakniti).

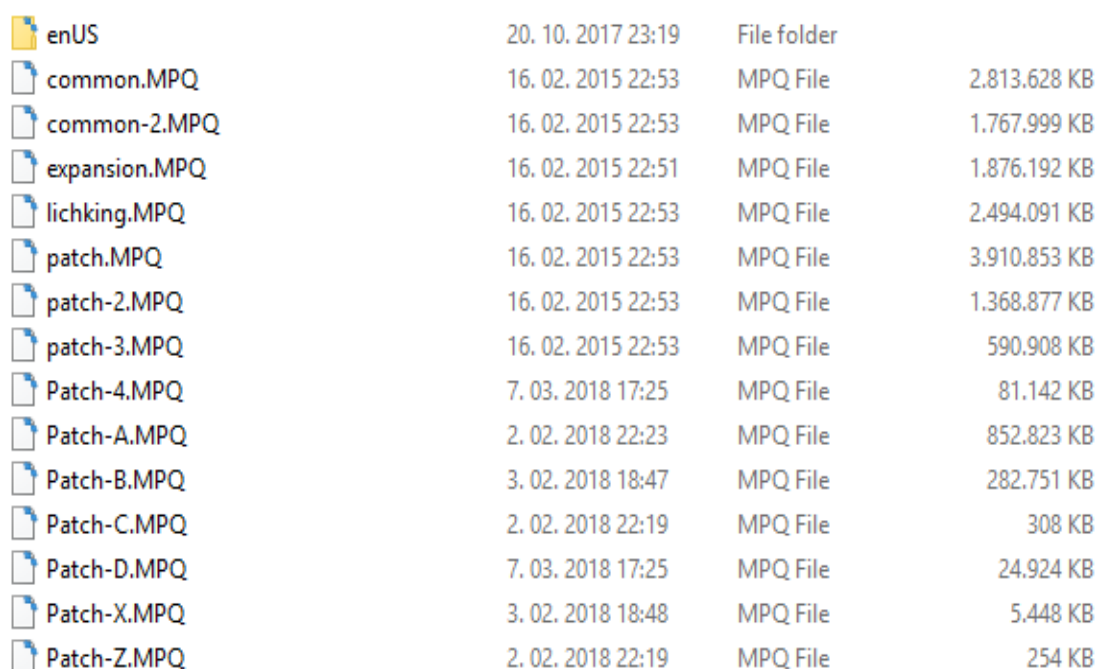
Za pridobitev zgornjih podatkov uporabljamo specifične ekstratorje, ki iz lokalne inštalacije igre World of Warcraft pridobijo potrebne podatke, ki jih potem preprosto prenesemo na strežnik World in ga tako lahko začnemo uporabljati. Premične mape in virtualne mape sicer niso obvezne za delovanje, vendar se brez njih v igri pogosto pojavlja veliko vrst napak. Ena izmed najbolj pogostih je, da NPC-ji padajo skozi tla, na katerih stojijo (saj klient samostojno ne obravnava njihove kolizije pravilno).

2.4 Distribucija posodobitev

Skozi razvoj igre World of Warcraft obstajata dva načina za distribucijo posodobitev. Prvi temelji na uporabi datotek MPQ, ki je format za arhiviranje datotek, ki ga podjetje Blizzard Entertainment uporablja v številnih svojih igrah. Format vsebuje možnost stiskanja, kriptiranja, segmentacijo, podpis (*signature*) in možnost shranjevanja več različic iste datoteke za večjezično podporo [7]. Ta način se je uporabljal od vsega začetka igre World of Warcraft, pa vse do razširitve Mists of Pandaria. Drugi pa uporablja novo podatkovno bazo Hotfixes, ki je bila implementirana od razširitve Warlords of Draenor dalje.

V nadaljevanju si bomo na kratko ogledali oba načina in jih primerjali med seboj.

2.4.1 Posodobitve preko MPQ-datotek



| | | | |
|---------------|--------------------|-------------|--------------|
| enUS | 20. 10. 2017 23:19 | File folder | |
| common.MPQ | 16. 02. 2015 22:53 | MPQ File | 2.813.628 KB |
| common-2.MPQ | 16. 02. 2015 22:53 | MPQ File | 1.767.999 KB |
| expansion.MPQ | 16. 02. 2015 22:51 | MPQ File | 1.876.192 KB |
| lichking.MPQ | 16. 02. 2015 22:53 | MPQ File | 2.494.091 KB |
| patch.MPQ | 16. 02. 2015 22:53 | MPQ File | 3.910.853 KB |
| patch-2.MPQ | 16. 02. 2015 22:53 | MPQ File | 1.368.877 KB |
| patch-3.MPQ | 16. 02. 2015 22:53 | MPQ File | 590.908 KB |
| Patch-4.MPQ | 7. 03. 2018 17:25 | MPQ File | 81.142 KB |
| Patch-A.MPQ | 2. 02. 2018 22:23 | MPQ File | 852.823 KB |
| Patch-B.MPQ | 3. 02. 2018 18:47 | MPQ File | 282.751 KB |
| Patch-C.MPQ | 2. 02. 2018 22:19 | MPQ File | 308 KB |
| Patch-D.MPQ | 7. 03. 2018 17:25 | MPQ File | 24.924 KB |
| Patch-X.MPQ | 3. 02. 2018 18:48 | MPQ File | 5.448 KB |
| Patch-Z.MPQ | 2. 02. 2018 22:19 | MPQ File | 254 KB |

Slika 2.8: Izgled posodobitev v MPQ-formatu.

MPQ-format je prvi način, ki ga je igra uporabljala za distribucijo posodobitev. Same posodobitve so zapakirane znotraj datotek MPQ (slika 2.8). V večini primerov so to nove teksture, modeli in spremenjene podatkovne baze, shranjene v DBC-datotekah. Zaradi tega so se številne datoteke lahko večkrat pojavile znotraj različnih MPQ-datotek, vendar je igra zastavljena tako, da prebere le tisto datoteko, ki je shranjena v najnovejši MPQ-datoteki. Vrstni red branja MPQ datotek je povsem navadno po abecednem redu. Tudi njihova imena so omejena, morajo vsebovati predpono »Patch-« in potem ena sama številka ali črka, kar pomeni, da je število MPQ datotek, ki jih lahko dodamo v igro omejeno. Torej, če imamo datoteko spell.dbc shranjeno v datoteki Patch-A.mpq in potem še v datoteki Patch-S.mpq, bo igra prebrala zgolj spell.dbc iz datoteke Patch-S.mpq. Seveda ima ta način distribuiranja posodobitev veliko problemov, in sicer:

- Podvajanje datotek, kar brez potrebe poveča velikost igre.
- Kadarkoli naredimo kakšno spremembo, moramo vsem igralcem dostaviti novo MPQ-datoteko, sicer ne bodo videli teh sprememb.
- Igralci lahko pobrišejo določeno MPQ-datoteko, kar jim lahko da določene prednosti pred drugimi igralci. Na primer, če datoteka Patch-M.mpq vsebuje varovano hišo in odstranimo to MPQ-datoteko, bo hiša nevidna za nas v igri in se lahko premaknemo na lokacijo, kjer vemo da ta hiša stoji. Nato dodamo MPQ-datoteko nazaj v našega klienta in tako uspešno vstopimo v hišo, ne da bi morali izpolniti pogoje za vstop.
- Omejeno število MPQ-datotek, ki jih igra lahko prebere. Načeloma jih do sedaj še nihče ni potreboval več, vendar če bi tudi v razširitvi Legion ali celo v prihodnosti še vedno uporabljali MPQ-datoteke, bi slej ko prej dosegli to omejitev.

2.4.2 Posodobitve preko podatkovne baze Hotfixes

- achievement
- achievement_locale
- anim_kit
- area_group_member
- area_poi
- area_poi_state
- area_table
- area_table_locale
- area_trigger
- armor_location
- artifact
- artifact_appearance
- artifact_appearance_locale
- artifact_appearance_set
- artifact_appearance_set_locale
- artifact_category
- artifact_locale
- artifact_power
- artifact_power_link
- artifact_power_rank
- artifact_quest_xp
- auction_house
- auction_house_locale
- bank_bag_slot_prices
- banned_addons
- barber_shop_style
- barber_shop_style_locale
- battle_pet_breed_quality
- battle_pet_breed_state
- battle_pet_species
- battle_pet_species_locale
- battle_pet_species_state
- battlemaster_list
- battlemaster_list_locale
- broadcast_text
- broadcast_text_locale
- char_sections
- char_start_outfit
- char_titles
- char_titles_locale
- chat_channels
- chat_channels_locale
- chr_classes
- chr_classes_locale
- chr_classes_x_power_types
- chr_races
- chr_races_locale
- chr_specialization
- chr_specialization_locale
- chr_upgrade_bucket
- chr_upgrade_bucket_spell
- chr_upgrade_tier
- cinematic_camera
- cinematic_sequences
- creature
- creature_difficulty
- creature_display_info
- creature_display_info_extra
- creature_family
- creature_family_locale
- creature_model_data
- creature_type
- creature_type_locale
- criteria
- criteria_tree
- criteria_tree_locale
- currency_types
- currency_types_locale
- curve
- curve_point
- destructible_model_data
- difficulty
- difficulty_locale
- dungeon_encounter
- dungeon_encounter_locale
- durability_costs
- durability_quality
- emotes
- emotes_text
- emotes_text_locale
- emotes_text_sound
- faction
- faction_locale
- faction_template
- gameobject_display_info
- gameobjects
- gameobjects_locale
- garr_ability
- garr_ability_locale
- garr_building
- garr_building_locale
- garr_building_plot_inst
- garr_class_spec
- garr_class_spec_locale
- garr_follower
- garr_follower_locale
- garr_follower_x_ability
- garr_plot
- garr_plot_building
- garr_plot_instance
- garr_plot_instance_locale
- garr_plot_locale
- garr_site_level
- garr_site_level_plot_inst
- gem_properties
- glyph_bindable_spell
- glyph_properties
- glyph_required_spec
- guild_color_background
- guild_color_border
- guild_color_emblem
- guild_perk_spells
- heirloom
- heirloom_locale
- holidays
- hotfix_data
- import_price_armor
- import_price_quality
- import_price_shield
- import_price_weapon
- item
- item_appearance
- item_armor_quality
- item_armor_shield
- item_armor_total
- item_bag_family
- item_bag_family_locale
- item_bonus
- item_bonus_list_level_delta
- item_bonus_tree_node
- item_child_equipment
- item_class
- item_class_locale
- item_currency_cost
- item_damage_ammo
- item_damage_one_hand
- item_damage_one_hand_caster
- item_damage_two_hand
- item_damage_two_hand_caster
- item_disenchant_loot
- item_effect
- item_extended_cost
- item_limit_category
- item_limit_category_locale
- item_modified_appearance
- item_price_base
- item_random_properties
- item_random_properties_locale
- item_random_suffix
- item_random_suffix_locale
- item_search_name
- item_search_name_locale
- item_set
- item_set_locale
- item_set_spell
- item_sparse
- item_sparse_locale
- item_spec
- item_spec_override
- item_upgrade
- item_x_bonus_tree
- key_chain
- lfg_dungeons
- lfg_dungeons_locale
- light
- liquid_type
- liquid_type_locale
- lock
- mail_template
- mail_template_locale
- map
- map_difficulty
- map_difficulty_locale
- map_locale
- modifier_tree
- mount
- mount_capability
- mount_locale
- mount_type_x_capability
- movie
- name_gen
- name_gen_locale
- names_profanity
- names_reserved
- names_reserved_locale
- override_spell_data
- phase
- phase_x_phase_group
- player_condition
- player_condition_locale
- power_display
- power_type
- pvp_difficulty
- quest_faction_reward
- quest_money_reward
- quest_package_item
- quest_sort
- quest_sort_locale
- quest_v2
- quest_xp
- rand_prop_points
- ruleset_item_upgrade
- scaling_stat_distribution
- scenario
- scenario_locale
- scenario_step
- scenario_step_locale
- scene_script
- scene_script_package
- skill_line
- skill_line_ability
- skill_line_locale
- skill_race_class_info
- sound_kit
- sound_kit_locale
- specialization_spells
- specialization_spells_locale
- spell
- spell_aura_options
- spell_aura_restrictions
- spell_cast_times
- spell_casting_requirements
- spell_categories
- spell_category
- spell_category_locale
- spell_class_options
- spell_cooldowns
- spell_duration
- spell_effect
- spell_effect_group_size
- spell_effect_scaling
- spell_equipped_items
- spell_focus_object
- spell_focus_object_locale
- spell_interrupts
- spell_item_enchantment
- spell_item_enchantment_condition
- spell_item_enchantment_locale
- spell_learn_spell
- spell_levels
- spell_locale
- spell_misc
- spell_power
- spell_power_difficulty
- spell_procs_per_minute
- spell_procs_per_minute_mod
- spell_radius
- spell_range
- spell_range_locale
- spell_reagents
- spell_scaling
- spell_shapeshift
- spell_shapeshift_form
- spell_shapeshift_form_locale
- spell_target_restrictions
- spell_totems
- spell_x_spell_visual
- summon_properties
- tact_key
- talent
- talent_locale
- taxi_nodes
- taxi_nodes_locale
- taxi_path
- taxi_path_node
- totem_category
- totem_category_locale
- toy
- toy_locale
- transport_animation
- transport_rotation
- unit_power_bar
- unit_power_bar_locale
- updates
- updates_include
- vehicle
- vehicle_seat
- wmo_area_table

Slika 2.9: Seznam tabel v podatkovni bazi Hotfixes.

Posodobitve preko podatkovne baze Hotfixes je novi način za distribucijo posodobitev. Ta način odpravi vse pomanjkljivosti načina posodobitev preko MPQ-datotek. Dejansko gre le za novo podatkovno bazo, ki se nahaja na samem strežniku. Ta vsebuje eno tabelo za vsako DBC-datoteko (slika 2.9). Dejansko razvijalci potem lahko dodajajo nove objekte v podatkovno bazo Hotfixes, namesto v DBC datoteke. Strežnik potem vse nove vpise v to podatkovno bazo sam prenese k igralcem med tem, ko igrajo igro. Oglejmo si prednosti uporabe podatkovne baze Hotfix:

- Ni nam treba več ročno prenašati datotek k igralcem, kar odpravi možnost, da bi igralci namerno odstranjevali datoteke.
- Ne uporabljamo več MPQ-datotek, torej ni več podvajanja.
- Spremembe lahko delamo veliko bolj pogosto, saj nam ni treba prenesti nobenih datotek k igralcem.
- Hotfixes je MySQL-podatkovna baza, kar nam omogoča izvajanje veliko bolj zahtevnih poizvedb kot v primeru DBC-datotek. Najbolj se to odraža, kadar želimo na hiter način spremeniti veliko število podatkov s pomočjo avtomatizacije.

Moramo se pa zavedati, da podatkovna baza Hotfix nadomesti zgolj posodobitve nad datotekami DBC, kar pomeni, da druge spremembe še vedno moramo ročno spraviti do igralcev, vendar je takšnih sprememb zelo malo v primerjavi.

2.5 Podatkovne baze MySQL za strežniške podatke

Zadnji del arhitekture TrinityCore pa so seveda podatkovne baze, v katerih hranimo vse podatke povezane s strežnikoma Auth in World. Podatki so razdeljeni v 3 (4 v novejših razširitvah) podatkovne baze, in sicer Auth, Characters in World (ter Hotfixes – glej poglavje 2.4.2). V nadaljevanju si bomo na kratko ogledali vsako izmed podatkovnih baz ter kakšne vrste podatkov se v njih shranjujejo. Tabel je zelo veliko, zato se bomo omejili zgolj na najbolj pomembne.

2.5.1 Podatkovna baza Auth

- account
- account_access
- account_banned
- autobroadcast R
- ip2nation
- ip2nationCountries
- ip_banned
- logs
- realmcharacters
- rbac_linked_permissions
- rbac_account_permissions
- rbac_default_permissions
- rbac_permissions
- realmlist
- uptime

Slika 2.10: Tabele v bazi Auth.

Kot lahko vidimo, je podatkovna baza Auth najmanjša, saj vsebuje zgolj 19 tabel (slika 2.10) v razširitvi Wrath of the Lich King, v razširitvi Legion pa vsebuje 29 tabel. Na sliki niso prikazane tabele, ki so uporabljene za samodejne posodobitve podatkovne baze. Ta baza se uporablja pretežno za shranjevanje podatkov o uporabniških računih in njihovih pravicah.

Gesla uporabnikov so seveda kriptirana, in sicer po formuli:

```
SHA1(CONCAT(UPPER(@username),':',UPPER(@password)))
```

To pomeni, da skupaj združimo uporabniško ime in geslo, med njih vstavimo še dvopičje ter vse skupaj kriptiramo z algoritmom SHA128. Vsebuje pa še tabele, ki določijo, kakšne pravice potrebuje nek uporabniški račun za uporabo določenih pooblaščenih ukazov. Nazadnje vsebuje še podatke o vseh kanalih, ki jih gostimo na strežniku.

2.5.2 Podatkovna baza Characters

- account_data
- account_instance_times
- account_tutorial
- addons
- arena_team
- arena_team_member
- auctionhouse
- banned_addons
- battleground_deserters
- bugreport
- calendar_events
- calendar_invites
- channels
- character_account_data
- character_achievement
- character_achievement_progress
- character_action
- character_arena_stats
- character_aura
- character_banned
- character_battleground_data
- character_battleground_random
- character_declinedname
- character_equipmentsets
- character_fishingsteps
- character_gifts
- character_glyphs
- character_homebind
- character_instance
- character_inventory
- character_pet
- character_pet_declinedname
- character_queststatus
- character_queststatus_daily
- character_queststatus_monthly
- character_queststatus_rewarded
- character_queststatus_seasonal
- character_queststatus_weekly
- character_reputation
- character_skills
- character_social
- character_spell
- character_spell_cooldown
- character_stats
- character_talent
- characters
- corpse
- creature_respawn I
- game_event_condition_save
- game_event_save
- gameobject_respawn I
- gm_subsurveys
- gm_surveys
- gm_ticket
- group_instance
- group_member
- groups
- guild
- guild_bank_eventlog
- guild_bank_item
- guild_bank_right
- guild_bank_tab
- guild_eventlog
- guild_member
- guild_member withdraw
- guild_rank
- instance
- instance_reset
- item_instance
- item_loot_items
- item_loot_money
- item_refund_instance
- item_soulbound_trade_data
- lag_reports
- lfg_data
- mail
- mail_items
- pet_aura
- pet_spell
- pet_spell_cooldown
- petition
- petition_sign
- pool_quest_save
- pvpstats_battlegrounds
- pvpstats_players
- reserved_name R
- warden_action
- worldstates

Slika 2.11: Tabele v bazi Characters.

Podatkovna baza Characters v razširitvi Wrath of the Lich King vsebuje 92 tabel (slika 2.11), v Legion pa 117 tabel. Na sliki zopet niso vključene tabele za samodejne posodobitve. Večina tabel se navezuje na igralniške like od vsakega igralca, kot so recimo njihovi atributi, vsi predmeti, inventura, sposobnosti, dosežki ipd. Prav tako so tukaj shranjeni podatki o skupinah (*guild*), torej podatki o vseh članih, predmetih, ki jih imajo shranjene v skupinski inventuri, do katere imajo dostop višji člani skupine.

2.5.3 Podatkovna baza World

- access_requirement R
- achievement_criteria_data R
- achievement_dbc
- achievement_reward R
- achievement_reward_locale R
- areatrigger_involvedrelation R
- areatrigger_scripts T
- areatrigger_tavern R
- areatrigger_teleport R
- battleground_template T
- battlemaster_entry
- command R T
- conditions R T
- creature
- creature_addon
- creature_classlevelstats T
- creature_equip_template
- creature_formations
- creature_loot_template R
- creature_model_info
- creature_onkill_reputation R
- creature_questender R
- creature_questitem
- creature_queststarter R
- creature_summon_groups R
- creature_template R
- creature_template_addon
- creature_template_locale R
- creature_template_movement
- creature_text R
- creature_text_locale R-
- disables R
- disenchant_loot_template R
- event_scripts R
- exploration_basexp T
- fishing_loot_template R
- game_event
- game_event_arena_seasons
- game_event_battleground_holiday
- game_event_condition
- game_event_creature
- game_event_creature_quest
- game_event_gameobject
- game_event_gameobject_quest
- game_event_model_equip
- game_event_npcflag
- game_event_npc_vendor
- game_event_pool
- game_event_prerequisite
- game_event_quest_condition
- game_event_seasonal_questrelation
- game_teleR
- game_weather
- gameobject
- gameobject_loot_template R
- gameobject_questender R
- gameobject_questitem
- gameobject_queststarter R
- gameobject_scripts
- gameobject_template
- gameobject_template_addon
- gameobject_template_locale
- gossip_menu R
- gossip_menu_option R T
- gossip_menu_option_action
- gossip_menu_option_locale R
- graveyard_zone R
- instance_encounters
- instance_template
- item_enchantment_template R
- item_loot_template R
- item_set_names R
- item_set_names_locale R
- item_template
- item_template_locale R
- lfg_dungeon_rewards R
- lfg_entrances
- linked_respawn R
- mail_level_reward R
- mail_loot_template R
- milling_loot_template R
- npc_spellclick_spells R
- npc_text
- npc_text_locale R
- npc_trainer R
- npc_vendor R
- outdoorpvp_template
- page_text R
- page_text_locale R
- pet_levelstats T
- pet_name_generation T
- phase_area
- pickpocketing_loot_template R
- player_classlevelstats T
- player_factionchange_achievement
- player_factionchange_items
- player_factionchange_reputations
- player_factionchange_spells
- player_levelstats T
- player_xp_for_level T
- playercreateinfo T
- playercreateinfo_action T
- playercreateinfo_item
- playercreateinfo_skills
- playercreateinfo_spell_custom
- points_of_interest R
- points_of_interest_locale R
- pool_creature
- pool_gameobject
- pool_pool
- pool_quest
- pool_template
- prospecting_loot_templateR
- quest_details
- quest_greeting
- quest_mail_sender
- quest_objectives
- quest_offer_reward
- quest_poi R
- quest_poi_points R
- quest_request_items
- quest_template R
- quest_template_addon
- quest_template_locale R
- reference_loot_template R
- reputation_reward_rate R
- reputation_spillover_template R
- skill_discovery_template R
- scene_template R
- script_waypoint T
- skill_discovery_template R
- skill_extra_item_template R
- skill_fishing_base_level R
- skinning_loot_template R
- smart_scripts R
- spawn_group
- spawn_group_template
- spell_area R
- spell_bonus_dataR T
- spell_custom_attrT
- spell_dbc T
- spell_enchant_proc_data T
- spell_group R T
- spell_group_stack_rules R T
- spell_learn_spell R T
- spell_linked_spell R T
- spell_loot_template R
- spell_pet_auras R T
- spell_proc R
- spell_ranks T
- spell_required R
- spell_script_names T
- spell_scripts R
- spell_target_position R
- spell_threat R T
- spelldifficulty_dbc T
- transports
- trinity_string R T
- vehicle_accessory T
- vehicle_template_accessory T
- version
- warden_checks
- waypoint_data R
- waypoint_scripts R
- waypoints

Slika 2.12: Tabele v bazi World.

Podatkovna baza World je največja izmed baz, vsebuje kar 175 tabel v razširitvi Wrath of the Lich King (slika 2.12) ter 209 v Legion. Na sliki tudi tukaj niso vključene tabele za samodejno posodobitev podatkovne baze. Baza World vključuje podatke o vseh objektih in pojavih, ki nastopajo v svetu.

V nadaljevanju si bomo ogledali nekaj najbolj pomembnih tabel, saj je vseh preveliko za bolj podroben pregled.

- Tabele s predpono areatrigger se uporabljajo za sproženje dogodkov, kadar nek igralec pride v točno določeno okolico. Ko se ta dogodek zgodi, se po navadi prične izvajati neka skripta.
- Tabele s predpono creature so uporabljene za definicijo vseh NPC-jev, njihovih lastnosti, pozicije v svetu, kako poteka interakcija z igralci (nasprotnik, prodajalec, nevtralen, itd.), kakšne predmete imajo. Lahko tudi določimo, kaj naj NPC reče ob različnih dogodkih, recimo, ko ga napademo ali premagamo.
- Tabele s predpono gameobject so podobne kot tabele s predpono creature, vendar gre tukaj za razne objekte namesto NPC. Lahko določimo, kako poteka interakcija z objekti (pasti, skrinje z zakladi, vrata, itd.), pogoje za interakcijo z objekti, njihova lokacija v svetu in tako dalje.
- Tabele s predpono playercreateinfo določajo lastnosti za vsako kombinacijo rase in razreda ob kreiranju novega igralniškega lika. Določimo lahko njihove začetne predmete, sposobnosti, lokacijo v svetu, kjer začnejo igro, odnos do drugih frakcij ipd.
- Tabele s predpono quest definirajo dogodivščine. V teh tabelah definiramo potovanje od začetka pa do konca. Določimo, kje se potovanje začne, kakšne so zahteve za pričetek, kaj moramo narediti, da dokončamo potovanje, kakšne so nagrade za uspeh, kakšne so sankcije, če nam ne uspe, in še veliko drugih funkcionalnosti, povezanih s potovanji.
- Tabele s predpono spell nadgradi funkcionalnost lokalnih podatkovnih baz DBC (poglavje 2.1.3). V tabelah lahko določimo interakcijo med sposobnostmi ter drugimi tabelami, ki smo si jih ogledali zgoraj. Tako lahko na primer povežemo sposobnost s potovanjem, na primer, ko uporabimo določeno sposobnost na določenem kraju, lahko napredujemo potovanje.

Na kratko smo si ogledali funkcionalnosti, ki jih ponujajo najbolj pomembne tabele v podatkovni bazi World. Nekatero izmed njih si bomo še bolj podrobno ogledali v nadaljevanju, ko si bomo ogledali praktične primere implementacije novih objektov.

3 Praktični primeri

V nadaljevanju si bomo ogledali praktične primere nad projektom TrinityCore, ki so rezultat mojega lastnega dela pri projektu. Ogledali si bomo, kako izgleda potek dodajanja novih funkcionalnosti v samem strežniku, nato pa še celoten postopek dodajanja novega objekta NPC v podatkovno bazo.

3.1 Implementacija nove funkcionalnosti v TrinityCore strežniku

TrinityCore že sam po sebi ponuja veliko funkcionalnosti, vendar pa še vseeno včasih odkrijemo napake ali pomanjkljivosti. V tem primeru se lahko odločimo, da sami napišemo skripto v programskem jeziku C++ in jo dodamo v strežnik TrinityCore.

Skripte načeloma ločimo v 2 vrsti, in sicer:

- Nadgradnja; v tem primeru dejansko ne začnemo pisati neke nove skripte, temveč se osredotočimo na spremembo obstoječe skripte, ki se že nahaja znotraj strežnika. Ta način se pogosto uporablja, kadar strežnik že vsebuje neki osnutek funkcionalnosti, ki jo želimo implementirati. Tako ne bi bilo smiselno podvajati kode, če lahko obstoječo zgolj nadgradimo.
- Nove funkcionalnosti, kadar želimo dodati funkcionalnost, ki nasploh še ni prisotna v strežniku. Skripte, ki jih v celoti napišemo sami, se nahajajo v direktoriju `src\server\scripts\Custom`. To je zelo pomembno, ker od razširitve Warlords of Draenor dalje lahko vse skripte, ki se nahajajo v tem direktoriju nastavimo, da se samostojno prevedejo med samim izvajanjem strežnika World, kar močno pospeši in olajša razvoj.

Določene skripte razvijalci objavljajo tudi na spletu, kar omogoča, da lahko določene splošne funkcionalnosti, ki jih večina ljudi želi imeti implementirane, lahko preprosto prenesejo ter vstavijo v strežnik brez znanja programiranja.

```

20 class death_system : public ItemScript
21 {
22 public:
23     death_system() : ItemScript("death_system") { }
24
25     Player *victim = NULL;
26     bool OnUse(Player* player, Item* item, SpellCastTargets const& /*spell target(s)*/) override
27     {
28         if (!player->GetSelectedPlayer()) {
29             player->GetSession()->SendAreaTriggerMessage("You must select a player.");
30         } else {
31             victim = player->GetSelectedPlayer();
32             if (player->HasAura(DEATH_STUN_SPELLID)) {
33                 player->GetSession()->SendAreaTriggerMessage("You must not be incapacitated.");
34             } else if (player->GetDistance(victim) > 5.0f) {
35                 player->GetSession()->SendAreaTriggerMessage("You are too far away.");
36             } else {
37                 ClearGossipMenuFor(player); //
38                 AddGossipItemFor(player, 0, "Execute", GOSSIP_SENDER_MAIN, DEATH_EXECUTE);
39                 AddGossipItemFor(player, 0, "Beat Up", GOSSIP_SENDER_MAIN, DEATH_BEAT_UP);
40                 AddGossipItemFor(player, 0, "Free", GOSSIP_SENDER_MAIN, DEATH_FREE);
41                 SendGossipMenuFor(player, 1, item->GetGUID());
42             }
43         }
44         return false;
45     }
46
47     void OnGossipSelect(Player* player, Item* /*item*/, uint32 /*sender*/, uint32 action) override
48     {
49         player->PlayerTalkClass->ClearMenus();
50         if (!victim)
51             return;
52         if (!victim->HasAura(DEATH_STUN_SPELLID)) {
53             player->GetSession()->SendAreaTriggerMessage("You must select a downed target.");
54             return;
55         }
56         std::string message;
57
58         switch (action)
59         {
60             case DEATH_EXECUTE:
61                 victim->TeleportTo(209, 419, 767, 10, 0);
62                 message = "has killed " + victim->GetName() + ".";
63                 player->TextEmote(message);
64                 sWorld->BanCharacter(player->GetName(), "-1", "Character RP Dead", "Server");
65                 break;
66             case DEATH_BEAT_UP:
67                 if (victim->HasAura(DEATH_STUN_SPELLID)) {
68                     victim->RemoveAura(DEATH_STUN_SPELLID);
69                 }
70                 victim->AddAura(BEAT_UP_SPELLID, victim);
71                 message = "has beaten up " + victim->GetName() + ".";
72                 player->TextEmote(message);
73                 break;
74             case DEATH_FREE:
75                 victim->RemoveAura(DEATH_STUN_SPELLID);
76                 message = "has freed " + victim->GetName() + ".";

```

Slika 3.1: Primer skripte.

Zgoraj je priložen del skripte, ki prikazuje implementacijo nove funkcionalnosti (slika 3.1). Skripto smo napisali za določen strežnik, ki je imel zelo specifično željo, ki močno odstopa od splošne funkcionalnosti igre World of Warcraft, in sicer implementira način, da lahko igralci, ki premagajo drugega igralca za večno zaplenijo igralcev lik, kar pomeni, da ga premagani igralec ne more več igrati in mora kreirati nov igralniški lik.

Sedaj pa si oglejmo še, kako so organizirane same skripte.

- Funkcije znotraj skript so dejansko dogodki, ki jih proži strežnik World med izvajanjem. Na primer, funkcija *OnUse* se sproži, kadar nek igralec desno klikne (uporabi) neki predmet. Seveda moramo v naši skripti nastaviti pogoje, kadar želimo, da se naša skripta dejansko izvede, sicer se bi izvedla vedno, kadar katerikoli igralec uporabi katerikoli predmet, kar pa po navadi ni to, kar želimo.
- Večina funkcij vrača vrednost tipa boolean (torej true ali false), kar sporoči strežniku World, če se je skripta uspešno zaključila ali ne. Če se ni, bo strežnik že sam zabeležil napako v datoteko za beleženje napak.

3.2 Dodajanje novih objektov v podatkovni bazi

Največ dela kot razvijalec za TrinityCore je delo s podatkovno bazo MySQL, in sicer dodajanje in spreminjanje objektov. V tem poglavju si bomo ogledali celoten postopek za dodajanje novega objekta v podatkovno bazo. Seveda pa bomo vzeli malce bolj zahteven primer, da bomo pokrili čim več nastavitev, ki nastopajo pri kreiranju novih objektov.

Ogledali si bomo tabele *creature_template*, *creature_template_outfit*, *creature_equip_template*, *creature_template_addon* in *creature_text*.

3.2.1 Tabela *creature_template*

| creature_template (80x1) | | | | | | | | | | |
|--------------------------|------------------------|----------------------|--------------------|----------------|--------------------|--------------|--------------|----------------|-------------|----------------|
| entry | difficulty_entry_1 | difficulty_entry_2 | difficulty_entry_3 | KillCredit1 | KillCredit2 | modelid1 | modelid2 | modelid3 | modelid4 | |
| 600.120 | 0 | 0 | 0 | 0 | 0 | -600.120 | -600.120 | -600.120 | -600.120 | |
| name | femaleName | subname | TitleAlt | IconName | gossip_menu_id | minlevel | | | | |
| Councilor Victoria | Councilor Victoria | The Council of Libra | (NULL) | (NULL) | 0 | 80 | | | | |
| maxlevel | HealthScalingExpansion | RequiredExpansion | VignetteID | faction | npcflag | speed_walk | speed_run | scale | rank | dmgschool |
| 80 | 5 | 0 | 0 | 31 | 0 | 1,1 | 2 | 1 | 3 | 0 |
| BaseAttackTime | RangeAttackTime | BaseVariance | RangeVariance | unit_class | unit_flags | unit_flags2 | | | | |
| 1.500 | 2.000 | 1 | 1 | 2 | 32.832 | 2.048 | | | | |
| unit_flags3 | dynamicflags | family | trainer_class | type | type_flags | type_flags2 | lootid | pickpocketloot | skinloot | resistance1 |
| 0 | 0 | 0 | 0 | 0 | 108 | 0 | 0 | 0 | 0 | 0 |
| resistance2 | resistance3 | resistance4 | resistance5 | resistance6 | spell1 | spell2 | spell3 | spell4 | spell5 | spell6 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| spell6 | spell7 | spell8 | VehicleId | mingold | maxgold | AIName | MovementType | InhabitType | HoverHeight | HealthModifier |
| 0 | 0 | 0 | 0 | 0 | 0 | | 0 | 3 | 1 | 1.300 |
| HealthModifierExtra | ManaModifier | ManaModifierExtra | ArmorModifier | DamageModifier | ExperienceModifier | RacialLeader | | | | |
| 1 | 200 | 1 | 1 | 1 | 1 | 0 | | | | |
| movementId | RegenHealth | mechanic_immune_mask | flags_extra | ScriptName | VerifiedBuild | | | | | |
| 204 | 1 | 1 | 0 | | 16.048 | | | | | |

Slika 3.2: Vnos v *creature_template*.

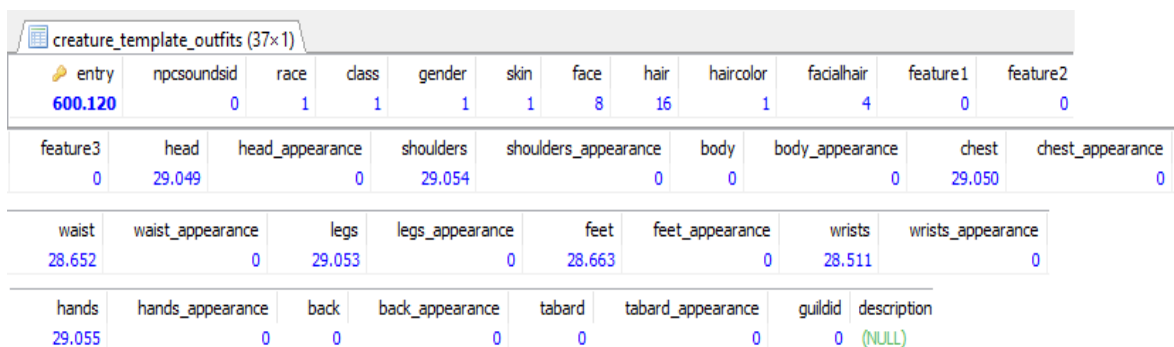
Tabela `creature_template` je dejanska tabela, kjer bomo ustvarili NPC, vse ostale bodo le pomožne tabele za dodajanje specifičnih lastnosti za ustvarjen NPC. Na zgornji sliki (slika 3.2) lahko vidimo končen rezultat.

Sedaj pa si bomo ogledali, kaj pomenijo posamezne vrednosti. Spustili bomo določene nedefinirane in nepomembne lastnosti.

- `Entry`: unikatna identifikacijska številka za vsak NPC. Ta številka se uporablja kot referenca v drugih tabelah ter za pooblašene ukaze znotraj igre.
- `Modelid1-4`: za vsak NPC lahko določimo do največ 4 različne modele. Če želimo, da ima le en model, bo prvi imel vrednost zelenega modela, drugi pa bodo 0. V našem primeru je model negativna številka, to je funkcionalnost, ki jo doda skripta `DressNPCs`, ki močno poenostavi kreiranje novih modelov. Če je torej `modelid` pozitivna številka, gre za klasični model, če pa je negativna, pa gre za model kreiran s pomočjo skripte.
- `Name`, `femaleName`, `subName`: `Name` in `femaleName` določata ime NPC-ja znotraj igre, odvisno od spola. Spol določi izbrani model. `SubName` pa je besedilo, ki se prikaže pod imenom. Ponavadi tja napišemo ime frakcije, ki ji pripada.
- `Minlevel` in `maxlevel`: določita stopnjo (*level*), ki jo bo imel NPC znotraj igre. Stopnja bo naključna vrednost med `minlevel` in `maxlevel` in bo različna za vsako instanco NPC-ja.
- `Faction`: določi frakcijo, ki ji pripada NPC. V našem primeru je to 31, ki je nevtralna proti vsem drugim frakcijam.
- `Npcflags`: bitno polje, ki določi vloge NPC-ja, kot so recimo: trener, bankir, prodajalec, ipd.
- `Speed_walk` in `speed_run`: določata, kako hitro se NPC-premika, kadar hodi in kadar teče.
- `Scale`: določi velikost NPC-ja. Vrednost je lahko med 0.1 in 10. Privzeta vrednost 1 pomeni normalna velikost, kot je definirana v samem modelu.
- `BaseAttackTime` in `RangeAttackTime`: kako hitro lahko NPC-napade od blizu (na primer z mečem) in kako hitro od daleč (na primer z lokom). Vrednosti so v milisekundah.
- `Unit_class`: numerična vrednost, ki določi, kateremu razredu pripada NPC. Razredi so isti, kot jih lahko izberejo igralci (torej warrior, paladin, itd.).

- `Unit_flags`, `unit_flags2` in `unit_flags3`: bitna polja. Ta polja vsebujejo največ možnosti izmed vseh atributov v tej tabeli. Določajo stanje NPC-ja, recimo če ga lahko napademo, če lahko uporablja sposobnosti, če lahko plava, če proba zbežati, ko ima malo življenjskih točk in še veliko drugih.
- `Dynamic_flags`: podobno kot `unit_flags`, vendar gre za vizualne lastnosti, namesto funkcionalne lastnosti. Nastavimo lahko lastnosti, kot so, da NPC izgleda mrtev.
- `Family`: določi vrsto, ki ji pripada NPC, na primer: humanoid, demon, volk, pajek itd.
- `Type`: podobno kot `family`, vendar osredotočeno na sam tip NPC-ja, recimo: mehaničen, humanoid, velikan, zver itd.
- `Type_flags`: bitno polje, ki določa tip NPC-ja ter njegovo stanje, recimo: prestižno bitje, šef (*boss*), duh itd.
- `Lootid`: ključ do tabele `creature_loot_template`, kjer definiramo, katere predmete ima na sebi NPC in jih lahko igralec pobere, ko ga premaga.
- `Resistance1-6`: kako odporen je NPC proti posameznim vrstam napadov.
- `Spell1-8`: vsakemu NPC-ju lahko dodamo sposobnosti. Vsako polje od `spell1` do `spell8` lahko vsebuje eno sposobnost.
- `AIName`: določi privzeto obnašanje NPC-ja. Lahko izbiramo med 8 vrst obnašanj.
- `InhabitType`: na kakšni površini se lahko premika NPC. Izbiramo lahko med funkcijami voda (*water*), tla (*ground*), letenje (*flying*), zakoreninjen (*rooted*).
- `HoverHeight`: kako visoko NPC lebdi nad zemljo.
- `ScriptName`: ime skripte, ki je povezana z NPC-jem. S tem povežemo skripto iz strežnika World z določenim NPC-jem.

3.2.2 Tabela creature_template_outfits



| entry | npc_soundsid | race | class | gender | skin | face | hair | haircolor | facialhair | feature1 | feature2 | feature3 | head | head_appearance | shoulders | shoulders_appearance | body | body_appearance | chest | chest_appearance | waist | waist_appearance | legs | legs_appearance | feet | feet_appearance | wrists | wrists_appearance | hands | hands_appearance | back | back_appearance | tabard | tabard_appearance | guildid | description |
|---------|--------------|------|-------|--------|------|------|------|-----------|------------|----------|----------|----------|--------|-----------------|-----------|----------------------|------|-----------------|--------|------------------|--------|------------------|--------|-----------------|--------|-----------------|--------|-------------------|--------|------------------|------|-----------------|--------|-------------------|---------|-------------|
| 600.120 | 0 | 1 | 1 | 1 | 1 | 8 | 16 | 1 | 4 | 0 | 0 | 0 | 29.049 | 0 | 29.054 | 0 | 0 | 0 | 29.050 | 0 | 28.652 | 0 | 29.053 | 0 | 28.663 | 0 | 28.511 | 0 | 29.055 | 0 | 0 | 0 | 0 | 0 | 0 | (NULL) |

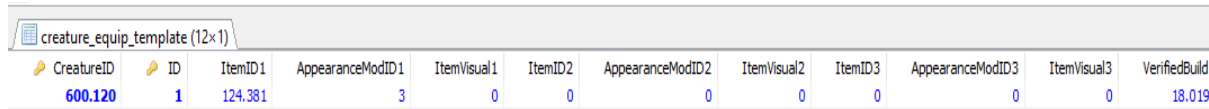
Slika 3.3: Vnos v creature_template_outfits.

Tabela creature_template_outfits je del skripte DressNPCs, ki močno poenostavi proces kreiranja novih modelov. Modele lahko kreiramo znotraj same podatkovne baze, namesto preko datotek DBC in DB2. Na sliki (slika 3.3) lahko vidimo končen rezultat.

Sedaj pa si oglejmo, kaj pomenijo posamezne vrednosti.

- Entry: unikaten identifikator za vsak model.
- Npc_soundsid: modelu lahko pripravimo želeni glas in zvočne efekte.
- Race, class, gender, face, hair, haircolor, feature1, feature2: določijo izgled modela. Lastnosti so popolnoma enake kot pri igralniških likih, zato je možno zelo preprosto za poljubnega igralca kreirati svoj model ter NPC.
- Preostale vrednosti pa se nanašajo na obleke, ki jih nosi model. Vsaka vrednost predstavlja referenco na predmet znotraj igre s tem ID-jem. Vsakemu delu obleke sledi še atribut appearance, ki določi izgled obleke (nekateri predmeti imajo na voljo več barv).
- Description: poljuben opis za model, da lažje najdemo to, kar iščemo.

3.2.3 Tabela creature_equip_template



| CreatureID | ID | ItemID1 | AppearanceModID1 | ItemVisual1 | ItemID2 | AppearanceModID2 | ItemVisual2 | ItemID3 | AppearanceModID3 | ItemVisual3 | VerifiedBuild |
|------------|----|---------|------------------|-------------|---------|------------------|-------------|---------|------------------|-------------|---------------|
| 600.120 | 1 | 124.381 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 18.019 |

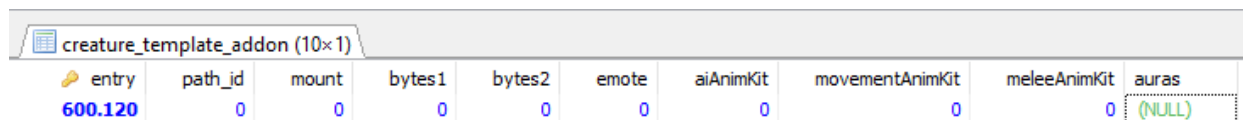
Slika 3.4: Vnos v tabelo creature_equip_template.

Tabela creature_equip_template je zelo preprosta, saj vsebuje zgolj podatke o orožjih, ki jih nosi posamezni NPC. Na sliki 3.4 lahko vidimo končen rezultat.

Oglejmo si kaj pomenijo posamezna polja:

- CreatureID: unikaten identifikator za vsak vnos.
- ID: NPC ima lahko več različnih naborov orožja. Če ima več naborov, bo igra izbrala naključnega znotraj igre.
- ItemID1-3: referenca na predmet znotraj igre, ki ga bo NPC nosil na sebi.
- AppearanceModID1-3 in ItemVisual1-3: določijo izgled predmeta, saj imajo nekateri predmeti več različnih izgledov (večinoma gre zgolj za različne barve istega predmeta).

3.2.4 Tabela creature_template_addon



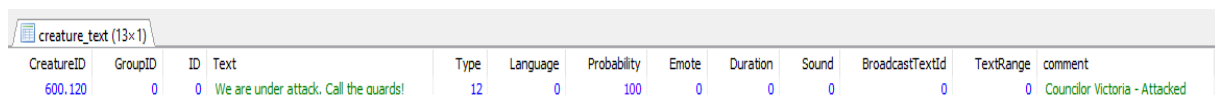
| entry | path_id | mount | bytes1 | bytes2 | emote | aiAnimKit | movementAnimKit | meleeAnimKit | auras |
|---------|---------|-------|--------|--------|-------|-----------|-----------------|--------------|--------|
| 600.120 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | (NULL) |

Slika 3.5: Vnos v tabelo creature_template_addon.

Tabela creature_template_addon ima vlogo, da določi obnašanje NPC-ja znotraj igre. Lahko nastavimo, katere sposobnosti ima, katero akcijo izvaja, katero bitje jezdi itd. Na sliki 3.5 lahko vidimo končen rezultat vnosa. Oglejmo si pomen posameznih atributov.

- Entry: unikaten identifikator za vsak vnos v tabelo.
- Mount: referenca na model, naj jezdi ta NPC.
- Bytes1: lahko določimo akcijo, ki naj jo izvaja ta NPC, na primer: sedenje na stolu, spanje itd.
- Bytes2: določimo, kako naj NPC drži orožje. Izbiramo lahko med tem, da NPC ne drži nobenega orožja, da NPC drži predmeta ItemID1 in ItemID2 iz tabele creature_equip_template ali pa da NPC drži le predmet ItemID3.
- Emote: določimo lahko akcijo, ki jo NPC izvaja periodično, na primer govorjenje, plesanje itd.
- Auras: Seznam sposobnosti, ki jih ima NPC.

3.2.5 Tabela creature_text



| CreatureID | GroupID | ID | Text | Type | Language | Probability | Emote | Duration | Sound | BroadcastTextId | TextRange | comment |
|------------|---------|----|---------------------------------------|------|----------|-------------|-------|----------|-------|-----------------|-----------|-------------------------------|
| 600.120 | 0 | 0 | We are under attack. Call the guards! | 12 | 0 | 100 | 0 | 0 | 0 | 0 | 0 | Councilor Victoria - Attacked |

Slika 3.6: Vnos v tabelo creature_text.

Tabela creature_text se uporablja v namen, da lahko NPC-ji ob določenih pogojih izvajajo interakcijo z igralci preko govora. Na sliki 3.6 vidimo končni rezultat. Oglejmo si pomen posameznih vrednosti.

- CreatureID: unikaten identifikator za vsak vnos v tabelo.
- GroupID: skupina besedil, če ima določen NPC več kot eno besedilo.
- ID: identifikator za vsako besedilo znotraj posamezne skupine besedil. Ob dogodku, se bo izbralo naključno besedilo znotraj skupine besedil.
- Text: Besedilo, ki ga bo NPC spregovoril.
- Type: določi, na kakšen način bo NPC spregovoril besedilo, recimo navaden govor, krik, šepetanje, itd.

- Probability: verjetnost, da bo izbrano to sporočilo.
- Emote: ko se sproži to sporočilo, lahko NPC izvede še določeno akcijo.
- Duration: kako dolgo naj bo vidno sporočilo.
- Sound: ob sporočilu se lahko predvaja določen zvok.
- TextRange: kako daleč stran lahko igralci vidijo to sporočilo.

3.2.6 Končen rezultat



Slika 3.7: Končni rezultat.

Na sliki 3.7 lahko vidimo, kako izgleda končen rezultat znotraj igre. Levi del slike prikazuje sam NPC, desni del pa prikazuje NPC, ko se sproži sporočilo, kadar igralec napade naš NPC. Kot lahko vidimo, imamo ogromno možnosti za prilagoditev NPC-jev in drugih objektov k našim željam in potrebam.

3.3 Izdelava spletne strani za TrinityCore

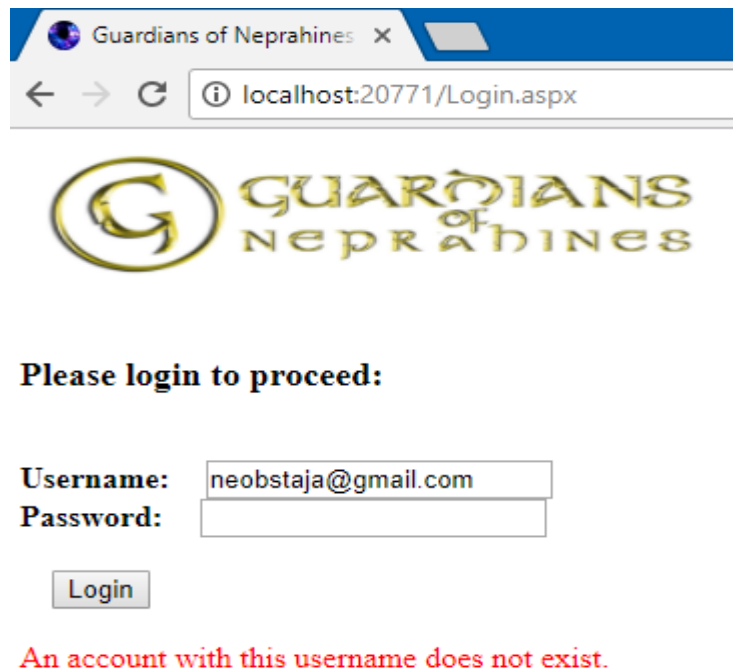
V nadaljevanju si bomo pogledali še spletno stran, ki sem jo naredil za RP-strežnike TrinityCore, ki se osredotočajo na uporabo administratorjev in upraviteljev igre za napredovanje v igri, namesto klasičnega načina napredovanja preko dokončanja dogodivščin in premagovanje nasprotnikov. V splošnem pooblašcene osebe organizirajo dogodke, ki se jih igralci lahko udeležijo in so na koncu nagrajeni, odvisno od tega, koliko so prispevali k uspehu dogodka. Prav tako pooblašcene osebe nagradijo tudi osebe izven dogodkov, če pozitivno vplivajo na preostale igralce z organiziranjem svojih manjših dogodkov.

TrinityCore sam po sebi ne ponuja nobene funkcionalnosti, ki bi omogočala nagrajevanje igralcev na RP strežnikih, zato sem se odločil, da kreiram spletno stran, kjer imajo dostop tako igralci, kot tudi pooblašcene osebe. Igralci lahko vidijo vse svoje igralniške like in lahko distribuirajo pridobljene točke v zelene attribute, pooblašcene osebe pa imajo dostop to administratorske strani, kjer lahko poljubnemu igralcu spremenijo attribute in jih lahko nagradijo.

Spletna stran je sestavljena modularno in omogoča zelo preprosto prilagoditev za potrebe uporabnika, kar vključuje podatke za povezavo s podatkovno bazo MySQL, katere attribute želi uporabnik prikazati na spletni strani. Omogoča tudi zelo preprost sistem za generiranje podatkov za tabelo stats_levels, kjer so definirane cene (atribut cost PT) za vsako stopnjo vsakega atributa, ter tudi koliko vsaka stopnja prispeva k splošni stopnji (atribut total power level) igralniškega lika. Spletna stran sestoji iz 4 strani, in sicer: stran za prijavo v sistem, uporabnikova nadzorna plošča (*user panel*), nadzorna plošča za upravljanje (*admin panel*) ter stran za ogled zgodovine. Vsako izmed njih si bomo ogledali v nadaljevanju. Pri izdelavi spletne strani sem si pomagal z literaturo, katera mi je pomagala pri pridobivanju znanja za povezovanje spletne strani s podatkovno bazo MySQL [8].

V prihodnje nameravam spletno stran še nadgraditi, kot prvo želim izboljšati izgled spletne strani, da bo izgledala bolj moderno. Želim pa dodati tudi nekaj novih funkcionalnosti, odvisno od tega, kaj bomo v praksi potrebovali, na primer možnost dodajanja predmetov igralcem.

3.3.1 Stran za prijavo v sistem



Slika 3.8: Prijava v sistem.

Na prvi strani imamo klasičen sistem za prijavo v sistem z enakim uporabniškim imenom in geslom, ki ga imamo za samo igro. Uporabniško ime je od razširitve Warlords of Draenor dalje elektronski naslov. Kot lahko vidimo na sliki (3.8) sistem pravilno zazna, če elektronski naslov ne obstaja ter tudi, če je kombinacija elektronskega naslova in gesla napačna. Če je prijava uspešna, potem v tej fazi v sejo shranimo podatke, ki jih bo spletna stran potrebovala v nadaljevanju, kot so na primer uporabnikove pravice. Po uspešni prijavi uporabnika preusmerimo na naslednjo stran, in sicer uporabnikova nadzorna plošča.

3.3.2 Uporabnikova nadzorna plošča

The screenshot shows a web browser window with the URL `localhost:20771/UserPanel.aspx`. The page features the logo for 'GUARDIANS OF NEPRAHINES' and a navigation bar with 'Select Character: Tester', 'Logout', and 'Admin Panel' buttons.

Key statistics displayed are:

- Total Power Level: 200
- Unspent Progression Tokens: 1850

There is an option to 'Select Stat: Spell Power' with an 'Upgrade!' button. A message states: 'You can buy this upgrade. (1850/450). This upgrade will increase your Power Level by 45.'

| Character | Health | Energy | Strength | Speed | Spell Power | Intelligence | Physical Defense | Magical Defense | Bonus PL | Silver |
|-----------|--------|--------|----------|-------|-------------|--------------|------------------|-----------------|----------|--------|
| Tester | 2 | 0 | 0 | 0 | 7 | 0 | 0 | 0 | 0 | 0 |

| Level | Cost (PT) | Cost Total (PT) | Power Level | Power Level Total |
|-------|-----------|-----------------|-------------|-------------------|
| 1 | 100 | 100 | 10 | 10 |
| 2 | 150 | 250 | 15 | 25 |
| 3 | 200 | 450 | 20 | 45 |
| 4 | 250 | 700 | 25 | 70 |
| 5 | 300 | 1000 | 30 | 100 |
| 6 | 350 | 1350 | 35 | 135 |
| 7 | 400 | 1750 | 40 | 175 |
| 8 | 450 | 2200 | 45 | 220 |
| 9 | 500 | 2700 | 50 | 270 |
| 10 | 550 | 3250 | 55 | 325 |
| 11 | 600 | 3850 | 60 | 385 |
| 12 | 650 | 4500 | 65 | 450 |
| 13 | 700 | 5200 | 70 | 520 |
| 14 | 750 | 5950 | 75 | 595 |
| 15 | 800 | 6750 | 80 | 675 |
| 16 | 850 | 7600 | 85 | 760 |
| 17 | 900 | 8500 | 90 | 850 |
| 18 | 950 | 9450 | 95 | 945 |
| 19 | 1000 | 10450 | 100 | 1045 |
| 20 | 1050 | 11500 | 105 | 1150 |
| 21 | 1100 | 12600 | 110 | 1260 |
| 22 | 1150 | 13750 | 115 | 1375 |
| 23 | 1200 | 14950 | 120 | 1495 |
| 24 | 1250 | 16200 | 125 | 1620 |
| 25 | 1300 | 17500 | 130 | 1750 |

Slika 3.9: Uporabnikova nadzorna plošča.

Uporabnikova nadzorna plošča je stran, kjer lahko igralci za vse svoje igralniške like pregledajo svoje atribute, ter lahko točke za napredovanje (*progression tokens*) tudi distribuirajo v atribute. Stran omogoča igralcu seznam vseh svojih igralniških likov, ter za trenutno izbranega v tabeli izpiše vrednosti vseh atributov, ter tudi splošno stopnjo in število točk, ki jih lahko distribuirajo. Na sliki je prikazan primer, kjer uporabnik namerava za igralniški lik Tester nadgraditi atribut Spell Power (slika 3.9). Pri tem primeru lahko vidimo, da ima igralec dovolj točk za nakup (1850), atribut pa stane 450 točk. Zato je uporabniku izpisano zeleno obvestilo, da lahko opravi nakup. Podobno bi se izpisalo tudi rdeče sporočilo, če uporabnik ne bi imel dovolj točk. V tem primeru je gumb za nakup onemogočen. V drugi tabeli so za vsako stopnjo prikazane cene (atribut cost PT), celotna cena od 1. stopnje do trenutne stopnje (atribut cost total PT), koliko se zviša splošna stopnja igralca, če kupi ta atribut (atribut power level) in še koliko je ta atribut do trenutne stopnje v celoti prispeval k splošni stopnji igralca (atribut power level total).

Nazadnje sta na strani na voljo še gumba za izpis, ki nas preusmeri nazaj na prvo stran ter gumb za dostop na stran za upravljanje, vendar je ta gumb na voljo le uporabnikom, ki imajo pooblašene pravice, sicer je neviden.

3.3.3 Stran za upravljanje

Select Character: Luana

Select Stat: Health

Select Action: Add

Amount: 2

Added 2 Health to character: Luana.

| Guid | Character | Health | Energy | Strength | Speed | Spell Power | Intelligence | Physical Defense | Magical Defense | Bonus PL | Silver | Unspent Tokens |
|------|-----------|--------|--------|----------|-------|-------------|--------------|------------------|-----------------|----------|--------|----------------|
| 28 | Luana | 9 | 16 | 4 | 8 | 20 | 15 | 8 | 12 | 0 | 5500 | 2500 |

| Guid | Character | Health | Energy | Strength | Speed | Spell Power | Intelligence | Physical Defense | Magical Defense | Bonus PL | Silver | Unspent Tokens |
|------|------------|--------|--------|----------|-------|-------------|--------------|------------------|-----------------|----------|--------|----------------|
| 131 | Ace | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2000 |
| 27 | Ace2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2000 |
| 136 | Achkarian | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2000 |
| 99 | Achkarian2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2000 |
| 104 | Adam | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2000 |
| 141 | Admin | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2500 | 3000 | |
| 107 | Aelaei | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2000 |
| 54 | Aendis | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2000 |
| 9 | Aion | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2000 |
| 117 | Aldurin | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2000 |
| 129 | Amilia | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2000 |
| 128 | Anne | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2000 |
| 106 | Arthanis | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2000 |
| 21 | Auxion | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2000 |
| 135 | Bobby | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2000 |
| 52 | Boby | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2000 |

Slika 3.10: Stran za upravljanje.

Naslednja stran, ki si jo bomo ogledali, je stran za upravljanje, do katere imajo dostop izključno pooblašeni uporabniki. Na tej strani lahko uporabnik v spodnji tabeli (slika 3.10) vidi vse igralniške like in vse njihove attribute. Uporabnik lahko nato izbere katerikoli lik, ki ga želi urejati. Ko izbere lik, se zgornja tabela osveži z aktualnimi podatki izbranega lika. Pooblašeni uporabnik nato izbere želeni atribut, želeno akcijo med dodaj, odštej in nastavi na (set to). Nazadnje določi še numerično vrednost in pritisne na gumb za izvedbo akcije. V zeleni barvi se nato izpiše obvestilo o izvedeni akciji, ki nam potrdi uspešno izvedbo akcije.

Na strani so na voljo še 3 gumbi, in sicer gumb za izpis, ki nas vrne na prvo stran, gumb za vrnitev na uporabnikovo ploščo in gumb za ogled zgodovine, ki se odpre v novem zavihku.

3.3.4 Stran za ogled zgodovine



Slika 3.11: Pregled zgodovine.

Zadnja stran pa je stran za ogled zgodovine. Ta stran je zelo preprosta, saj preprosto prikaže zgodovino vseh izvedenih akcij. Kot lahko vidimo na sliki 3.11, sta razvidna datum in čas izvedene akcije, kdo je izvedel akcijo ter seveda, kaj je ta akcija bila.

4 Sklepne ugotovitve

V diplomski nalogi je predstavljen celoten pregled arhitekture in delovanje projekta TrinityCore. Predstavljeni so tudi praktični primeri, ki prikažejo, kako izgleda potek dodajanja novih funkcionalnosti in dodajanja novih objektov. Ogleдали smo si tudi praktični izdelek – spletna stran za upravljanje z igralci na RP-strežnikih, kjer veljajo drugačna pravila kot na klasičnih strežnikih. To temo sem si izbral s ciljem, da predstavim projekt, ki služi kot celovito orodje za vpogled v delovanje MMORPG-iger, saj imamo popoln pogled v celotno izvorno kodo projekta.

Čeprav projekt TrinityCore omogoča ogromno funkcionalnosti in prilagoditev za lastne potrebe, pa ima tudi številne probleme. Že ob prikazu praktičnih primerov v podatkovni bazi MySQL je bilo razvidno, da je dodajanje novih objektov lahko precej dolgotrajno delo, saj je večina atributov shranjenih v numerični obliki, kar pa pomeni, da brez dokumentacije ne vemo, kaj pomeni posamezna vrednost. Dolgoročni cilj bi bila izdelava vmesnika, ki bi razvijalcem prikazal te vrednosti v obliki oznak, ki bi jih lažje razumeli. Prav tako je dokumentacija zaenkrat še nepopolna, nekatera polja v podatkovni bazi sploh še nimajo definiranih vrednosti, tako razvijalcem preostane le to, da slepo preizkušamo vrednosti, in skušamo ugotoviti kaj določena vrednost pomeni. Prav tako je problem tudi majhna skupina razvijalcev, ki sodelujejo na projektu TrinityCore, saj še preden lahko implementirajo funkcionalnosti iz trenutne razširitve, že pride ven naslednja, kar v praksi pomeni, da so iz vsake razširitve implementirane le najbolj pomembne funkcionalnosti, preostale pa so preprosto primorani zanemariti.

Literatura

- [1] Informacije o TrinityCore. Dostopno na: <https://www.trinitycore.org/>. [Dostopano 10. 3. 2018].
- [2] Kaj je emulacija/emulator. Dostopno na: <https://en.wikipedia.org/wiki/Emulator>. [Dostopano 10. 3. 2018].
- [3] Podatki o igri World of Warcraft. Dostopno na: <https://worldofwarcraft.com/en-us/>. [Dostopano 12. 3. 2018].
- [4] Simulacija epidemije v World of Warcraft. Dostopno na: https://en.wikipedia.org/wiki/Corrupted_Blood_incident. [Dostopano 12. 3. 2018].
- [5] Informacije o podatkovni bazi DBC. Dostopno na: <http://fox.wikis.com/wc.dll?Wiki~DatabaseContainer>. [Dostopano 15. 3. 2018].
- [6] Informacije o podatkovni bazi DB2. Dostopno na: https://en.wikipedia.org/wiki/IBM_Db2. [Dostopano 16. 3. 2018].
- [7] Podatki o formatu MPQ. Dostopno na: [https://en.wikipedia.org/wiki/MPQ_\(file_format\)](https://en.wikipedia.org/wiki/MPQ_(file_format)). [Dostopano 16. 3. 2018].
- [8] Imar Spaanjaars, *Beginning ASP.NET 4: in C# and VB*, Nizozemska, 2010, pogl. 13.
- [9] Leigh Achterbosch, Robyn Pierce in Gregory Simmons, "Massively multiplayer online role-playing games: the past, present, and future", *Computers in Entertainment (CIE) - Theoretical and Practical Computer Applications in Entertainment*, št. 5, 4. izdaja, 5. artikel, 2007.
- [10] Richard A. Bartle, *Designing Virtual Worlds*, ZDA, 2004, pogl. 2.