



**ČESKÉ VYSOKÉ
UČENÍ TECHNICKÉ
V PRAZE**

F3

**Fakulta elektrotechnická
Katedra počítačů**

Bakalářská práce

System pro predikci úspěšnosti a výpočtu skóre módu „Mythic+“ ve hře World of Warcraft

Datový model, sběr dat, logika systému a vizualizace dat

David Kula

Otevřená informatika

Květen 2021

Vedoucí práce: Ing. Jiří Šebek

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Kula** Jméno: **David** Osobní číslo: **474781**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra počítačů**
Studijní program: **Otevřená informatika**
Studijní obor: **Software**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

System pro predikci úspěšnosti a výpočtu skóre módu „mythic+“ ve hře World of Warcraft. – Datový model, sběr dat, logika systému a vizualizace dat

Název bakalářské práce anglicky:

System for predicting success and calculating the score of the "mythic+" mode in the game World of Warcraft. – Data model, data collection, system logic and visualisation of data

Pokyny pro vypracování:

1. Naprogramujte addon do hry World of Warcraft v jazyce Lua, sbírající relevantní data o hráčích pro ohodnocení hráče v dungeonech.
2. Sesbírejte data generovaná hrou, během dungeonu pro pozdější užití.
3. Vytvořte testovací sadu dat.
4. Analyzujte sesbíraná data, zvolte nejvhodnější atributy dat k predikci úspěchu a výpočtu skóre.
5. Implementujte business logiku systému a prezentační vrstvu, kde budou data vhodně vizualizována.
6. Otestujte, zjistěte relevanci daného rating systému. Otestujte systém vůči jiným systémům, které poskytují takové skóre.

Seznam doporučené literatury:

1. Angular. Angular [online]. 2021 [cit. 2021-02-09]. Dostupné z: <https://angular.io/docs>
2. WHITESTEAD II, James a Rick ROE. World of Warcraft Programming: A Guide and Reference for Creating WoW Addons. 2nd Edition. 10475 Crosspoint Boulevard, Indianapolis: Wiley Publishing, 2010. ISBN 978-0470481288.
3. Ludicrous Speed, LLC. "World of Warcraft Rankings for Mythic+ and Raid Progress." Raider.IO, 2017, raider.io/
4. Lua [online]. Rio de Janeiro, RJ - Brasil - 22451-900 - Cx. Postal: 38097, 2020 [cit. 2021-02-09]. Dostupné z: <https://www.lua.org/>

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Jiří Šebek, kabinet výuky informatiky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **20.02.2021**

Termín odevzdání bakalářské práce: **21.05.2021**

Platnost zadání bakalářské práce: **19.02.2023**

Ing. Jiří Šebek
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta

Poděkování / Prohlášení

Chtěl bych poděkovat Martinu Lukešovi, že se mnou spolupracoval na tomto systému, také vedoucímu Jiřímu Šebkovi, že nám umožnil pracovat na tomto tématu pod jeho vedením. Dále děkuji všem testerům, kteří nám pomohli otestovat všechny bizarní i nebizarní případy a strávili s námi ve hře i spoustu náročných a zdoluhavých chvil. V poslední řadě chci poděkovat své rodinně, za podporu při studiu a přítelkyni, která to se mnou celou dobu studia vydržela.

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací. V Praze dne 20.5.2021

Abstrakt / Abstract

Tato bakalářská práce se zaměřuje na implementaci systému hodnocení KeystonePro, který hodnotí hráče hry World of Warcraft v herním módu Mythic+ a predikuje úspěch týmů v tomto herním módu. Systém posbírání data o hráčích, sám je zpracuje a vygeneruje každému zaznamenanému hráči jeho skóre. V práci najdeme lehký přehled možných technologií pro vývoj systému, bližší pohled na implementaci sběru dat, frontendu a části byznys logiky a rozbor dat použitých k jeho uvedení do provozu. Na této práci bylo pracováno v jazyce Lua v editoru Visual Studio Code, ve frameworku Angular v také ve Visual Studio Code a v jazyce Python v IDE Pycharm.

Klíčová slova: World of Warcraft, Mythic Keystone, Mythic+ systém hodnocení, predikce úspěchu v dungeonu

This bachelor thesis focuses on implementation of ranking system KeystonePro, which ranks World of Warcraft players in game mode Mythic+ and predicts success of teams in this game mode. System collects data about players, processes them and generates each player a score. In thesis we may find a brief overview over usable technologies for development of the system, closer look on implementation of data collection, frontend and part of business logic and analysis of data used for system's commissioning. This thesis was developed in coding language Lua in Visual Studio Code editor, framework Angular also in Visual Studio Code editor and coding language Python in Pycharm IDE.

Keywords: World of Warcraft, Mythic Keystone, Mythic+ ranking system, prediction of succes in dungeon

Obsah /

1 Úvod	1
1.1 Motivace.....	1
2 Rešerše	2
2.1 Herní mód Mythic+	2
2.2 Mythic+ ratingový systém Raider.io.....	4
2.3 Addony a pluginy v hrách	5
2.3.1 Addony ve hře World of Warcraft	7
3 Související práce	8
4 Analýza	9
4.1 Mythic+ ratingový systém KeystonePro	9
4.1.1 Herní addon	10
4.1.2 Desktopová aplikace na posílání dat na server .	11
4.1.3 Serverová část	11
4.1.4 Databáze	11
4.1.5 Frontend - webová aplikace.....	11
4.2 Algoritmus Predikce	11
4.2.1 Perceptron	12
4.2.2 Support vector machine .	12
4.2.3 Adaboost	12
4.3 Metody generace dat	13
4.3.1 SMOTE	14
4.4 Sbíraná data	14
4.4.1 Smrti/Deaths	15
4.4.2 Celkové léčení/Total Healing Done.....	15
4.4.3 Celkové přeléče- ní/Total Overhealing Done.....	16
4.4.4 Léčení v boji/Combat healing.....	16
4.4.5 Přeléčení v bo- ji/Combat overhealing...	16
4.4.6 Absorbované poškoze- ní/Absorbs done	17
4.4.7 Nepřerušená kouz- la/Uninterrupted spells	18
4.4.8 Udělené poškoze- ní/Damage Done.....	18
4.4.9 Efektivní poškoze- ní/Effective Damage Done.....	18
4.4.10 Utržené poškoze- ní/Damage taken	19
4.5 Technologie addonu	19
4.6 Technologie frontendu	20
4.6.1 Angular.....	20
4.6.2 React	20
4.6.3 Vue	21
4.6.4 Porovnání	22
4.6.5 Závěr	24
4.7 Technologie predikce	25
4.7.1 Python	25
4.7.2 Java	26
4.7.3 Závěr	26
4.8 Databáze	26
4.9 Verzování.....	26
4.9.1 Git	26
4.9.2 Závěr	27
5 Dataset	28
5.1 InterruptScore.....	28
5.2 DamageDoneScore	29
5.3 DeathsScore	29
5.4 DamageReceivedScore	29
5.5 FailureDamageScore	29
5.6 FailureDamageRatio	30
5.7 HealingScore	30
5.8 DungeonLevel	30
5.9 DungeonName	31
5.10 CompletionTimeRatio	31
5.11 Vizualizace dat.....	31
5.12 Posudek vzorků dat.....	37
5.12.1 Sanguine Depths 14 - InTime	37
5.12.2 Spires of Ascension 12 - deplete	38
6 Návrh a implementace	41
6.1 Herní addon	41
6.1.1 Důležité herní eventy	43
6.1.2 Ukládání dungeonu na disk	46
6.2 Frontend.....	47
6.2.1 Blizzard API a Oauth ...	47
6.2.2 Interakce s uživatelem a servisní vrstva.....	48

6.3 Implementace Algoritmů	
Predikce	51
6.4 Implementace výpočtu skóre ..	53
7 Testování	54
7.1 Testování addonu	54
7.1.1 Unit testy	54
7.1.2 Buggy v průběhu vývoje addonu	55
7.1.3 Uživatelské testování addonu	55
7.2 Testování frontendu	56
7.2.1 Uživatelské testování frontendu	57
7.3 Testování predikce a skóre	57
7.3.1 Skóre	57
7.3.2 Predikce	60
8 Instalace	62
8.1 Instalace addonu	62
8.2 Instalace uploaderu	62
9 Závěr	64
10 Future Work	65
A Přílohy	67
Literatura	68

Tabulky /

4.1. Délky časů na dokončení jednotlivých dungeonů Módu Mythic+	13
4.2. Statistiky repozitářů front-endových frameworků na GitHub	25
7.1. Predikce úspěchu známých spoluhráčů	60
7.2. Predikce úspěchu nejlepších hráčů systému	60
7.3. Predikce úspěchu nejhorších hráčů v systému	60

Kapitola 1

Úvod

1.1 Motivace

World of Warcraft [1] je populární MMORPG (massively multiplayer online role-playing game) vydávané společností Blizzard Entertainment [2], ve které si hráč založí postavu a poté spolu s postavami jiných hráčů plní různé výzvy proti počítači (PvE - player versus environment) nebo soupeří přímo proti postavám jiných hráčů (PvP - player versus player). V obou případech hra po hráčích vyžaduje jistou znalost herních mechanik, taktik a nadále taky dobrou koordinaci hráčů v rámci týmu, kde má většinou každý přidělenou různou úlohu. Není proto divu, že nalézt v této hře schopné spoluhráče a vytvořit tak kompetitivní skupinu, zvláště potom na vyšších úrovních hry, může být problémem.

V této práci se konkrétně zaměříme na herní mód Mythic+, který se řadí do kategorie PvE. Tento mód se hraje v pěti hráčích, kteří spolu kooperují a snaží se projít mapu ("Dungeon") v časovém limitu a zároveň mapu vyčistit od nepřátelských NPC (non player character). Mapy mají různé obtížnostní úrovně. Tento herní mód nemá žádný oficiální ratingový systém a nejpoužívanější fanouškovský systém Raider.io [3] je podle nás málo podrobný a dostatečně nevypovídající. Bez dobrého rankingového systému je těžké poznat, jestli jsou jednotliví hráči dostatečně kvalitní, pro zvládnutí dané obtížnostní úrovně a jestli pokus o její zvládnutí nezkaží. Proto jsme se rozhodli zkusit napsat vlastní rankingový systém, který bude sbírat data o hráčích přímo ze hry během těchto Mythic+ runů.

Kapitola 2

Rešerše

V této části blíže představím herní mód Mythic+ a herní mechaniky vyskytující se a související s ním, protože se celá práce věnuje právě jemu. Dále přiblížím jak funguje nejvyužívanější dosavadní ratingový systém Raider.io.

2.1 Herní mód Mythic+

Herní mód Mythic+ se hraje v pětičlenných týmech. Cílem Mythic+ módu je dokončit dungeon včas - čas na dokončení dungeonu je dán typem dungeonu a odpočítávání začne při spuštění módu. Dungeon je uzavřená herní oblast, ve které se nachází 2 typy NPC (non player character). Dungeon je dokončen právě tehdy, když jej skupina vyčistí od předem stanoveného počtu trash NPC a všech boss NPC. První typ je obyčejné NPC, mezi hráči WoW slangově označované jako trash. Tento typ NPC se vyskytuje běžně všude po celém dungeonu a funguje jako jakási výplň. Moderní trend je ovšem dělat trash komplikovanější a přidávat mu všelijaké mechaniky. Druhý typ NPC v dungeonu je klasický boss. Boss je speciální NPC, který vydrží mnohem více než trash, rozdává hráčům více poškození a má speciální mechaniky, na které je třeba dávat pozor. Příkladem takové mechaniky je může být třeba když boss vyčaruje oheň na různá místa okolo něj a hráči se potom tomu ohni musí vyhnout.

Tým hráčů sestává z jednoho tanka, jednoho healera a tří damage dealerů. Tanci nosí těžkou zbroj, a vydrží spoustu poškození. Jejich úloha spočívá k upoutání veškeré pozornosti na sebe, musí přimět veškeré nepřátelské NPC aby útočili přímo na něj a vytváří tak prostor pro ostatní členy skupiny. Healer (česky léčitel) se stará o to, aby nikdo v týmu neumřel. Používá různá léčivá kouzla, aby všichni zůstali co nejzdravější. Damage dealeri (doslova rozdavači poškození) jsou pracovní síla skupiny. Jejich údělem je co nejrychleji zabít všechny nepřátelské NPC, které útočí na tanka. Dávají typicky vysoké poškození, ale mají zase malou odolnost a po útoku nepřátelských NPC snadno umírají.

Dungeony v módu Mythic+ mají různé úrovně. Čím vyšší úroveň, tím více poškození nepřátelské NPC vydrží a rozdají. S rostoucí úrovní dále přibývají a také se stávají silnějšími speciální efekty zvané afixy. Afix je speciální efekt, který platí po celou dobu trvání dungeonu. Typický afix je například, že bossové vydrží o 30% více a rozdají o 30% více poškození, anebo po zabití trash NPC exploduje a zraní všechny hráče v okolí. Každý týden se stanoví čtveřice afixů, která potom bude platit ve všech Mythic+ dungeonech.



Obrázek 2.1. Grafický interface vkládání klíče do dungeonu. [4]

Úroveň Mythic+ dungeonu je stanovena klíčem. Hráč získá klíč úrovně 2, při dokončení dungeonu na úrovni Mythic. Při dokončení dungeonu Mythic+ v čase, úroveň klíče se zvýší podle toho, kolik času ještě zbývá do vypršení limitu. Pokud se dungeon nedokončí včas, úroveň klíče se sníží, ale nikdy ne níž, než na úroveň 2.

Na obrázku 2.1 můžeme vidět grafický interface, který se zobrazí ve hře, při vložení klíče do dungeonu. Úplně nahoře vidíme úroveň klíče. Dole pod grafickým znázorněním klíče vidíme název dungeonu - Theater of Pain. Pod názvem dungeonu vidíme čas, který má tým k dispozici na jeho dokončení. Pod časem vidíme 6 ikon. První dvě znázorňují bonus k poškození a k hit pointům (100% a 100%) nepřátelských NPC v dungeonu na dané úrovni a zbylé čtyři jsou ikonkami afixů, které budou platit po celou dobu dungeonu.

Pokaždé co jeden z hráčů umře, odečte se z časového limitu 5 sekund a hráč má možnost oživit se na checkpointu (místo, které slouží jako oživovací místo, poté co se k němu tým probouje).

Na obrázku 2.2 můžeme vidět interface Mythic+ dungeonu v průběhu. Úplně nahoře je zobrazeno jméno dungeonu - Theater of Pain. Pod názvem vidíme jeho úroveň - level 12 - a vedle ní jsou 4 ikonky jednotlivých afixů, které jsou aktivní tento týden. V tomto případě vidíme zleva Tyrannical, Bursting, Explosive a Prideful. Pod ikonami afixů se nachází zbývající čas a počet smrtí skupiny.



Obrázek 2.2. Interface Mythic+ v průběhu dungeonu. [4]

2.2 Mythic+ ratingový systém Raider.io

Pokud se dnes chce hráč připojit do Mythic+ skupiny na nějaký dungeon, leader dané skupiny jej bude většinou porovnávat s ostatními aplikanty podle jeho Raider.io [3] skóre. Raider.io čte z blizzard API dokončené dungeony hráčem a za každý typ dungeonu se podívá na jeho nejlepší běh/run tímto dungeonem. Na základě tohoto nejlepšího runu stanoví skóre za tento dungeon a potom sečte skóre za všechny typy dungeonů. Tento součet je poté výsledným Raider.io skórem. Dále se zobrazují i jednotlivá skóre za jednotlivé dungeony, takže ostatní hráči mohou vidět, které dungeony do výsledného celkového skóre přispívají nejvíce.

Raider.io má několik nedostatků. Hodnotí výkon jednoho hráče podle výsledku celého týmu. Jeden hráč v týmu může být výrazně slabší než ostatní hráči, ale nakonec získá stejné skóre za daný dungeon. Ve hře World of Warcraft se navíc poměrně často vyskytuje boosting, to je, že si jeden slabý hráč za herní měnu, nebo i za reálné peníze, koupí pomoc týmu velice kvalitních hráčů se špičkovým vybavením a ti jej daným dungeonem provedou. Dále aby měli hráči dobré skóre podle Raider.io, musí chodit všechny dungeony a jakmile hráč nějakou dobu nehrál jeden nebo dva typy dungeonů, jeho skóre jde dolů. Tohle může být problém právě proto, že klíč k dungeonu se pro hráče generuje náhodně. Hráč si sice může zkusit vyhledat někoho jiného, kdo klíč k chybějícímu dungeonu má, ale může mít problém s přijetím do skupiny, protože jeho skóre za tento dungeon je poměrně nízké.

Dungeon (Score: 1,537.1)	Level	Time	Affixes	Score	World	Region
Mists of Tirna Scithe	+17 *	00:24:54	[Icons]	197.7	81,910	49,359
The Necrotic Wake	+17 *	00:31:39	[Icons]	196.8	80,933	48,018
Halls of Atonement	+17 *	00:27:49	[Icons]	196.6	78,195	47,197
Theater of Pain	+17 *	00:33:36	[Icons]	196.4	47,509	29,302
Spire of Ascension	+17 *	00:35:34	[Icons]	196.3	76,108	44,756
De Other Side	+17 *	00:39:49	[Icons]	196.1	60,330	36,154
Plaguefall	+16 *	00:33:28	[Icons]	178.9	134,940	81,332
Sanguine Depths	+16 *	00:37:47	[Icons]	178.3	75,527	46,084

Obrázek 2.3. Skóre hráče na webu Raider.io[5]

Na obrázku 2.3 vidíme skóre postavy hráče zobrazené na webu Raider.io. Na každém řádku vidíme nejlepší run postavy hráče v jednom konkrétním dungeonu. Ve sloupci score vidíme skóre za každý z těchto dungeon runů a potom v hlavičce prvního sloupce vidíme skóre celkové, které je sumou sloupce Score.

2.3 Addony a pluginy v hrách

V dnešní době se dají k rozšíření her použít různé modifikace. Hráči si mohou pro zpestření zážitku z hraní nainstalovat poměrně často nové grafické modely, a tak si například ve populární hře GTA 5 můžete zkusit pilotovat obří křižník galaktického imperia (zobrazeno na obrázku 2.4), nebo stíhačku T-65 ze Star Wars. Další populární hra, kde jsou různé šílené modifikace velmi oblíbené, je například Elder Scrolls: Skyrim. Tato hra se odehrává ve středověkém fantasy světě plném draků, trollů, skřetů, elfů a jiných magických bytostí. Jednou z nejoblíbenějších modifikací je Really Useful Dragons (zobrazeno na obrázku 2.5), která všechny draky ve hře nahradí obřími lokomotivami, které vypadají jako mašina Tomáš z populárního dětského seriálu.



Obrázek 2.4. Modifikace křižníku do GTA V [6]

Modifikace v online hrách nebývají zas tak časté, protože by mohly některým hráčům poskytovat neférovou výhodu oproti jiným, kteří by si modifikace nenainstalovali. I přes to lze v některých hrách modifikovat vzhled různých objektů, například ve hře League of Legends existují programy, které změní vzhled vaší herní postavičky. Tuto změnu vidíte však pouze vy u sebe na počítači. Ještě méně z těchto her poté podporuje pluginy a addony či modifikace, které by hráčům usnadňovaly hraní hry, nebo jinak pomáhaly ve zvýšení výkonu hráče. Ve hře Counter Strike: Global Offensive si hráči často mění vzhled svého zaměřovače pro lepší míření, jiné herní modifikace jsou však často ilegální a pokud se prokáže jejich používání, může hráči hrozit i ban na hraní dané hry. Mezi takové nepovolené pluginy patří ve hře CS:GO například aim bot, který automaticky zaměřuje hráčovu zbraň na hlavy protihráčů, nebo wall hack, který umožňuje hráčům dívat se skrz zdi a získat tak neférovou výhodu nad protihráči. Protože dnešní hry často addony pomáhající hráčům v průběhu hry nepodporují, vznikly takzvané overlaye. Nejedná se přímo o kód, který by se spustil v rámci hry, ale je to samostatná aplikace, která se zobrazí přes hru a doplňuje tak herní interface například překrytím nevyužitých ploch uživatelského rozhraní hry a ukazuje tam dodatečné informace o hře.

Takovým příkladem může být například Facecheck pro hru League of Legends. Jedná se o overlay, který zobrazuje hráčům jaké itemy stavět na jejich herní postavu a poskytuje jiné dodatečné info během hry i mimo ni. Overlaye většinou neinteragují přímo s hrou, protože hry často neposkytují žádné rozhraní. Některé overlaye sbírají data z internetových API daných her a poté je zobrazují živě přímo ve hře.

Jelikož různé hry jsou vytvořeny podle různých technologií a poskytují různá rozhraní, techniky psaní addonů a modifikací do her se pro každou hru mohou lišit.

Široce modifikovatelná hra je v dnešní době například Minecraft. Jednou z více cest jak modifikovat hru Minecraft je například Minecraft Forge [7]. Minecraft Forge je open-source server, který umožňuje hráčům instalovat a spouštět modifikace pro Minecraft. Modifikace přes Forge jsou neoficiální a hra Minecraft k nim nenabízí žádnou podporu. Forge bylo vyvinuto za účelem zjednodušení kompatibility komunitních Minecraft modifikací se samotnou hrou Minecraft: Java Edition. [8] Forge umožňuje modifikaci do hloubky, což je například tvorba vlastních itemů, vlastního chování různých NPC nebo přidání kompletních industriálních strojů.

Modifikace pro Minecraft s Minecraft Forge se programují v jazyce Java a poté je stačí přidat do adresáře mods v kořenovém adresáři hry Minecraft. Hráč však musí mít Minecraft Forge nainstalovaný, aby mohl tyto modifikace správně spouštět a to přímo tu stejnou verzi, jako má verzi hry Minecraft na svém počítači. Minecraft Forge musí být také nainstalovaný i na serveru hry Minecraft.



Obrázek 2.5. Modifikace Really Useful Dragons do hry Elder Scrolls: Skyrim [9]

■ 2.3.1 Addony ve hře World of Warcraft

Addony ve hře World of Warcraft jsou úplně jiná věc. World of Warcraft totiž modifikace hry přímo podporuje a poskytuje k nim API. Hráč si může ve hře přímo spouštět vlastní skripty v jazyce Lua, ke kterému je zároveň knihovna funkcí přímo pro hru World of Warcraft, a zároveň má přístup k veškerému uživatelskému rozhraní generovanému hrou přímo z těchto skriptů. Může si je proto různě měnit, rozšiřovat a nahraovat vlastní modely celého uživatelského rozhraní. Ve skriptech se lze také napojit na různé herní eventy, které signalizují různé herní události. Hra také umožňuje přístup k jejím logům přes herní eventy a tím pádem si hráč v těchto skriptech může vykreslit veškeré informace a události, které ve hře probíhají. Skripty stačí umístit do složky Inteface/AddOns v kořenovém adresáři s hrou a World of Warcraft si je poté sám interpretuje a spustí.

Kapitola 3

Související práce

Spolu se mnou se tomuto tématu také věnoval Martin Lukeš ve svojí práci [10]. Martin pracoval na druhé části systému a ve své práci se věnuje přenosu dat mezi uživatelským počítačem a serverem, serverem a databází a serverem a frontendem systému. Dále se věnuje zpracování a perzistenci nasbíraných dat, tvorbě samotného skóre z těchto nasbíraných dat - tudíž návrhu hodnotícího systému. Ve své práci zpracovává návrh technologií pro přenos dat mezi komponentami systému, technologii persistence a návrh databázového modelu a technologii serverové části systému. Dále, protože plánujeme vytvořit plně funkční a použitelný systém, se Martin ve své části stará i o deploy/nasazení backendu i frontendu na konkrétní doménu.

Kapitola 4

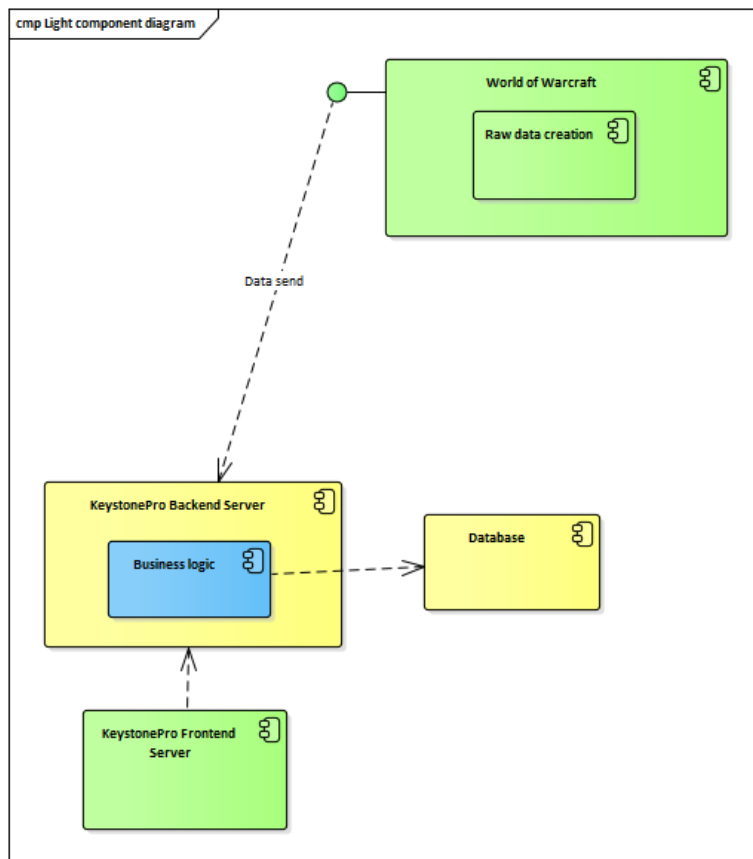
Analýza

V této části představím funkčnost našeho systému nazvaného KeystonePro. Dále přiblížím jaká data budeme z herního módu Mythic+ sbírat a co vlastně znamenají a nakonec proberu jednotlivé technologie hodící se k implementaci částí systému, které mám na starosti já (herní addon, frontend a predikce úspěchu na backendu).

4.1 Mythic+ ratingový systém KeystonePro

Idea našeho ratingového systému je monitorovat každého hráče během dungeonu, zaznamenávat kolik udělal různých chyb a kolik jednotlivá chyba ovlivnila průchod dungeonem, a také kolik přispíval vykonáváním své role k dokončení a urychlení postupu dungeonem, například kolik damage dealer rozdál poškození nepřátelským NPC a jak rychle je dokázal zneškodnit. Na základě těchto dat se poté spočítá hráčův rating. Dále pro ulehčení vytváření týmu pro dungeon, bude náš systém vytvářet predikci, zda-li daná pětice hráčů dokáže projít konkrétním dungeonem určité úrovně včas.

Na diagramu 4.1 můžeme vidět z jakých částí se bude náš systém skládat a jak budou na sebe napojené. Části implementované mnou jsou znázorněny zelenou barvou, části implementované Martinem jsou znázorněny žlutě a části na kterých jsme se podíleli oba jsou znázorněny modře.

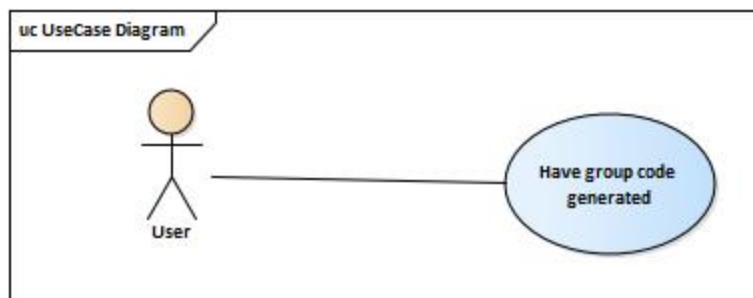


Obrázek 4.1. Obecný architektonický diagram[10]

■ 4.1.1 Herní addon

Herní addony bývají ve hře World of Warcraft velice běžné, většina hráčů je používá pro doplnění a rozšíření uživatelského rozhraní hry dle své potřeby. Pomocí addonu se dají monitorovat nejrůznější věci a potom například zobrazit formou grafického rozhraní uživateli. Náš addon má za úkol sbírat data ze hry, zaznamenat všechny události, které by potenciálně mohly ovlivnit čas dokončení dungeonu. Po dokončení dungeonu herní addon uloží data na disk k pozdějšímu posláni na serverovou část systému. Dále bude uživatel moci příkazem vygenerovat kód skupiny, který si potom může zkopírovat do schránky a následně vložit do frontendu systému k vygenerování predikce pro skupinu pěti hráčů.

Komunikaci s hráčem/uživatelem znázorňuje tento usecase diagram 4.2.



Obrázek 4.2. Use case diagram pro Addon

4.1.2 Desktopová aplikace na posílání dat na server

Původně jsme chtěli, aby addon nejen sbíral data, ale také byl sám schopen je rovnou odeslat na serverovou část. Blizzard tohle bohužel nepodporuje z bezpečnostních důvodů. Není možné cokoli posílat pryč v rámci addonů a kódu interpretovaného interpreterem ve hře World of Warcraft.[11]

Pro tento účel je tedy použita desktopová aplikace, která po manuálním zapnutí vyhledá data uložená na disku addonem a pošle je na náš server.

4.1.3 Serverová část

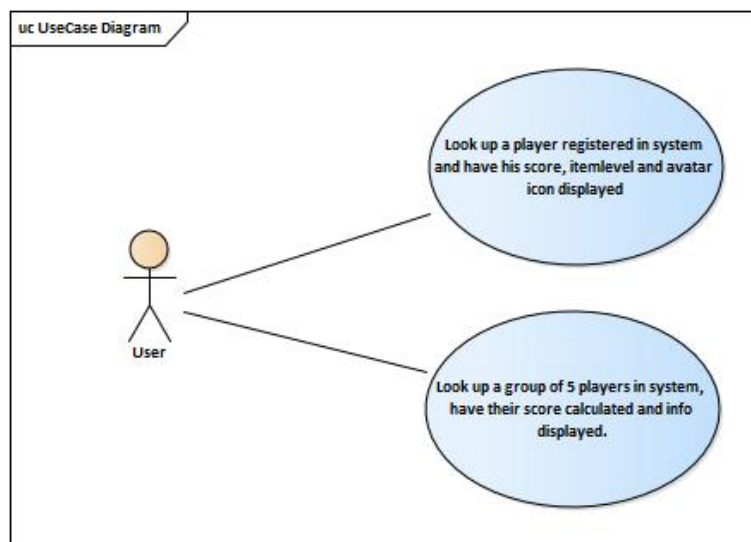
Na serveru bude probíhat veškerá manipulace s nasbíranými daty a vypočítání skóre jednotlivých hráčů a také samotná predikce úspěchu týmu v dungeonu. Server bude přijímat data nasbíraná addonem a odeslaná desktopovou aplikací a ukládat je do databáze, bude vyřizovat požadavky uživatelů zprostředkované frontendem systému a bude s nimi skrz frontend komunikovat.

4.1.4 Databáze

Do databáze se budou ukládat data nasbíraná addonem pro pozdější výpočet skóre nebo pro predikci.

4.1.5 Frontend - webová aplikace

Frontend bude sloužit ke komunikaci systému s uživateli systému. Budou si skrze něj moci zobrazit skóre jednotlivých postav a generovat predikce úspěchu pro pětičlenné skupiny hráčů. Frontend bude dostupný formou webové aplikace na url adrese, aby se uživatel mohl připojit pohodlně odkudkoliv a aby si nemusel instalovat víc softwaru, než je nutné. Interakci uživatele s frontendem můžeme vidět na tomto diagramu 4.3.



Obrázek 4.3. Usecase diagram Frontendu

4.2 Algoritmus Predikce

Cílem algoritmu predikce je predikovat, zda pětice hráčů uspěje v daném dungeonu dané úrovně. Jedná se o binární klasifikaci se dvěma hodnotami cílové proměnné: 1 = skupina uspěje a projde dungeon včas, -1 = skupina neuspěje a nedokončí dungeon

včas. Protože budeme mít k dispozici addon, který nám bude generovat záznamy o proběhlých dungeon runech, budeme moci tímto sestavit trénovací sadu, tudíž budeme používat supervizované učení.

■ 4.2.1 Perceptron

Perceptron je lineární algoritmus pro strojové učení na binární klasifikační úlohy. Může být považován za jeden z prvních a nejjednodušších typů umělých neuronových sítí. Zcela jistě se nejedná o “hluboké” učení, ale je to důležitý stavební kámen. Stejně jako logistická regrese, může se rychle naučit lineární separaci v prostoru vlastností pro klasifikační úlohy dvou tříd, ačkoliv narozdíl od logistické regrese se učí za pomoci stochastického sestupu gradientu optimalizačního algoritmu a nepředvídá kalibrované pravděpodobnosti. [12]

Jedná se bohužel o velice primitivní algoritmus, který nebude konvergovat, pokud nejsou data lineárně separabilní a tím pádem nedokáže vyřešit například XOR problém. Perceptron jde rozšířit tak, že dokáže klasifikovat i některá lineárně neseperabilní data.

■ 4.2.2 Support vector machine

Support vector machine algoritmus je velice podobný perceptronu. Také se snaží nalézt podprostor, kterým lineárně separuje data na dvě skupiny. Těchto podprostorů může být však nekonečně mnoho a každý z nich může poté v reálné predikci rozdělovat reálná data jinak. Support vector machine algoritmus se proto snaží najít nejlepší z těchto dělicích podprostorů. Stejně jako u perceptronu má také bohužel problém s lineárně neseperabilními daty. Existují však také metody jak tento problém překonat. Jednou z takových metod je použití kernelové funkce, která zvedne vstupních dimenzi dat a zobrazí je jako lineárně separabilní ve vyšší dimenzi. Kernelových funkcí je více druhů.

- RBF Kernel - RBF kernely jsou nejvíce obecná forma kernelizace a je to jeden z nejrozšířenějších kernelů díky svojí podobnosti s Gaussovým rozdělením. RBF kernel funkce pro dva body X_1 a X_2 počítá podobnost, nebo také jak blízko si navzájem jsou.

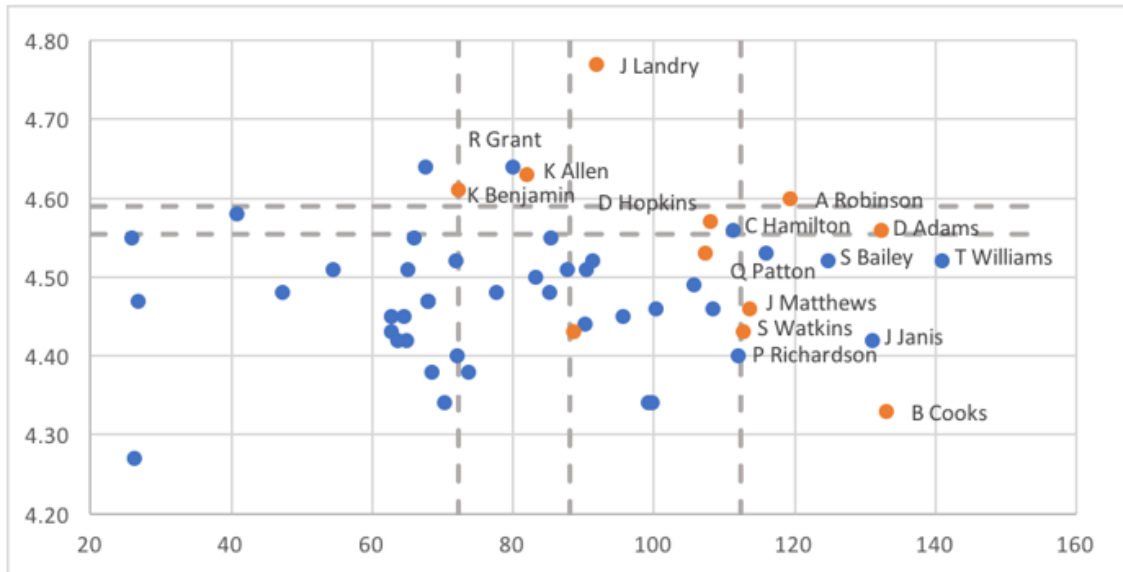
[13]

■ 4.2.3 Adaboost

Adaboost klasifikátor je meta-estimator, který začne fitovat klasifikátor na původním data setu a poté fituje dodatečné kopie toho klasifikátoru na stejném datasetu, který má upravené váhy u špatně klasifikovaných dat tak, aby se tyto kopie více soustředily na těžší vzorky dat ke klasifikaci. [14]

Adaboost kombinuje několik slabých klasifikátorů do jednoho silného klasifikátoru. Jako slabé klasifikátory se často používají jednoduché rozhodovací stromy, kterým se v tomto případě říká “decision stumps”. Každý z těchto stumpů má potom svůj rozhodovací threshold (práh) a klasifikuje jednotlivé vzorky dat podle toho, jestli leží pod nebo nad prahem. Výsledné rozhodnutí silného klasifikátoru je potom váženým hlasováním / lineární kombinací rozhodnutí těchto slabých klasifikátorů.

Na obrázku 4.4 můžeme vidět rozdělení dat pěti stumpy - slabými klasifikátory.



Obrázek 4.4. Klasifikace dat adaboostem [15]

4.3 Metody generace dat

Pro získání jednoho vzorku dat je třeba dokončit jeden dungeon Mythic+. Každý dungeon trvá jinak dlouho, průměrná délka času na jejich dokončení včas je 36 minut 52,5 sekund. Čas na dokončení dungeonu se nemění s jeho úrovní. Kromě běhu samotným dungeonem je třeba nejdřív sestavit tým hráčů přes herní systém LFG (looking for group), nebo se předem domluvit s jinými hráči. Jedna skupina se dle mých zkušeností skládá zhruba 5-10 minut, záleží na denní době a úrovni dungeonu, který skládáte. Po sestavení skupiny je třeba, aby všichni hráči dorazili na místo, což může zabrat zhruba 2-3 minuty. Když si tým v dungeonu nevede moc dobře a vzniknou pochybnosti, nebo se prokáže, že nedokáže dungeon dokončit včas, někteří hráči se rozhodnou opustit hru v průběhu dungeonu a dungeon se poté vůbec nedokončí, tudíž nedojde k posbírání vzorku. Když vezmeme tyto věci v potaz, sesbírání jednoho vzorku trvá hrubým odhadem 46 minut, s tím, že veškeré úsilí může přijít vniveč, pokud někdo opustí hru během dungeonu. Z tohoto odhadu lze usoudit, že nasbírání datasetu může být problematické, pokud náš systém najednou nezačne používat spousta hráčů.

Dungeon	Timer
De Other Side	43:00
Halls of Atonement	31:00
Mists of Tirna Scithe	30:00
Necrotic Wake	36:00
Plaguefall	38:00
Sanguine Depths	41:00
Spires of Ascension	39:00
Theater of Pain	37:00

Tabulka 4.1. Čas na dokončení jednotlivých dungeonů. [16]

4.3.1 SMOTE

Synthetic Minority Over-sampling Technique (SMOTE) a Modified-SMOTE jsou dvě techniky generující syntetická data. Jednoduše řečeno SMOTE vezme data pointy minoritní třídy a vytvoří nové data pointy ležící mezi dvěma nejbližšími data pointy spojenými rovnou čarou.

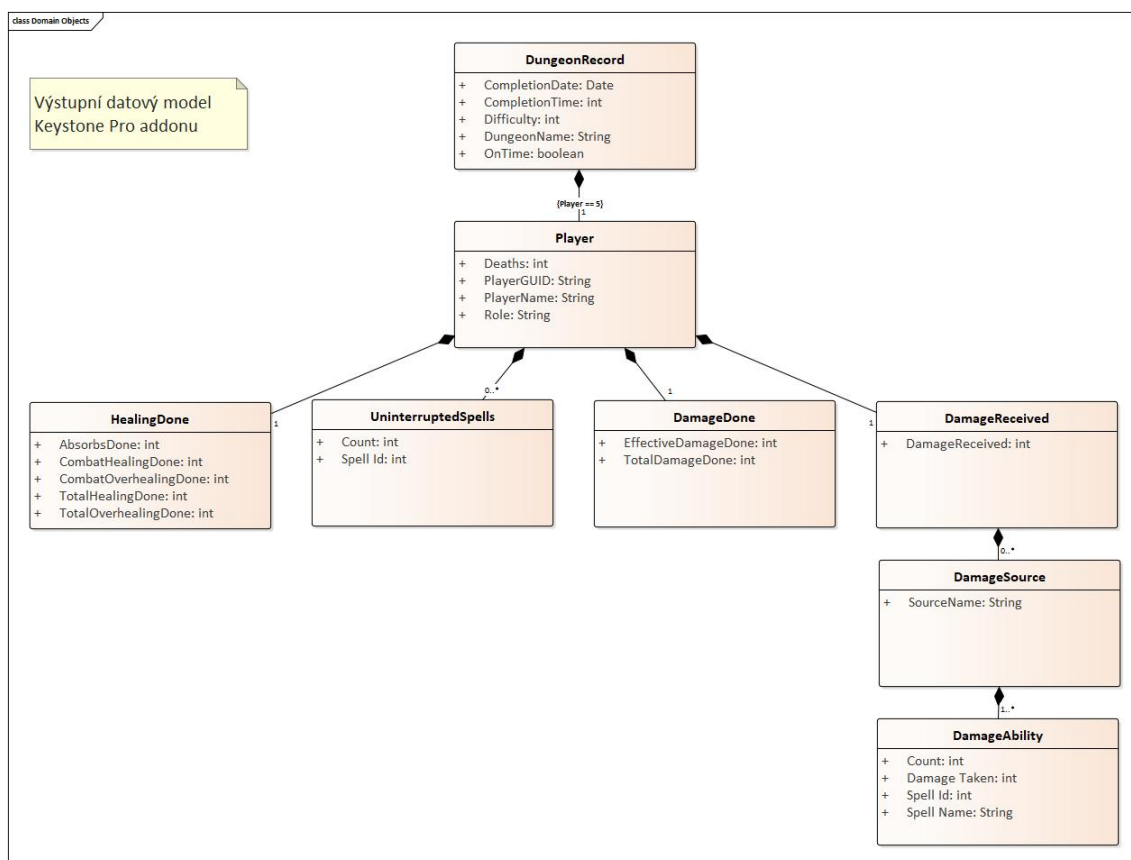
Algoritmus spočítá vzdálenost mezi dvěma data pointy v prostoru atributů/featur, a násobí tuto vzdálenost náhodným číslem mezi 0 a 1. Poté umístí nový data point na násobenou vzdálenost od jednoho z data pointů použitých pro výpočet téhle vzdálenosti. [17]

SMOTE pomáhá při problémem s undersamplingem, tedy když je jedna ze tříd v menšinovém nevyváženém poměru v ostatních třídách. SMOTE poté může zlepšit rozpoznatelnost minoritní třídy generováním nových umělých dat a zvýšit tak její reprezentaci v datasetu. [18]

Tato metoda také nese svá rizika, protože může vnést do nového obohaceného datasetu bias, případně posílit už existující bias, a také přichází i problémy s pochopením skutečné podoby dat v reálném světě. [17]

4.4 Sbíraná data

V této podkapitole se zaměříme na to, jaká data chceme sbírat o uživateli, proč a jak souvisí s Mythic+ herním módem a jaký mohou mít vliv na úspěch a neúspěch skupiny, jaký dávají smysl a jakými způsoby si je můžeme vyložit.



Obrázek 4.5. Datová struktura záznamu dungeonu

Na obrázku 4.5 můžeme vidět formát dat, který se chystáme sbírat ze hry a který potom budeme posílat dále ke zpracování.

4.4.1 Smrti/Deaths

Dle principů hry hráč by skoro nikdy neměl umřít jinak, než svou vlastní chybou (například stojí tam, kde by neměl, nebo se nedokázal vyhnout něčemu nebezpečnému včas). Smrti proto vypovídají o očividných chybách hráčů. Smrt hráče také přímo ovlivňuje úspěšnost týmu v dungeonu, protože po smrti hráče se odečítá 5 sekund ze zbývajících času na dokončení. Smrt hráče ovlivňuje úspěšnost týmu i nepřímo, protože umřeli tank nebo healer, tak to velice často vede ke smrti ostatních členů party, protože v případě smrti tanka se všechna nepřátelská NPC rozběhnou na ostatní členy party, kteří nemají potřebnou výzbroj a vybavení, aby jejich útok přežili, a nebo v případě smrti healera tank brzy podlehne útokům NPC a ty se poté opět rozběhnou na zbytek týmu. V případě smrti damage dealera není situace tak špatná, nicméně chvíli trvá, než se hráč s touto rolí oživí a doběhne zpátky do boje, a protože v týmu potom chybí jeho poškození, tak boj trvá déle. V případě, že umře v boji s bossem, nemůže se ani samovolně oživit a musí počkat, dokud neumře zbytek týmu, nebo tým v početní nevýhodě bosse nezdolá, což bude pochopitelně trvat o mnoho déle, protože boss vydrží většinou opravdu hodně. Na obrázku 4.6 můžeme vidět hráče, který je mrtvý. Umřel při boji s bossem.



Obrázek 4.6. Hráč ležící mrtvý v boji s bossem [4]

4.4.2 Celkové léčení/Total Healing Done

Primární role healerů je dávat pozor, kdykoliv dostane někdo poškození a jeho životy (hit points/health) klesnou pod maximální hodnotu a co nejrychleji tohoto hráče vyléčit svými schopnostmi. Tímto vzniká healing done, který monitorujeme, a celkový healing, který proběhl během celého dungeonu ukládáme jako TotalHealingDone. Léčení/healing však nemá žádný efekt, když je léčen hráč s plnými životy, takže pokud všichni dodržují taktiky jak mají a tank má dobré vybavení, může mít i dobrý healer malý healing done a to sice proto, že to zkrátka není vůbec potřeba. To však neznamená, že healing není důležitý. V některých momentech je skupina pod velkým tlakem a opravdu záleží na

čisté léčivé síle healera. Také čím lepší healer, tím rychleji může skupina postupovat dále, protože je schopna vyvolat boj s více skupinami nepřátel.

Na obrázku 4.7 vidíme hráče, který má na první části 7795 hit pointů a čaruje léčivé kouzlo. Na druhé části obrázku ho již vyléčil a vyléčil se na 13551 hit pointů.



Obrázek 4.7. Hráč se léčí/healuje [4]

4.4.3 Celkové přeléčení/Total Overhealing Done

Když healer použije léčící schopnost na někoho, kdo už má plné životy, nebo pokud použije mnohem silnější schopnost, než je třeba, vzniká overhealing/přeléčení. Nejde o žádný negativní efekt, ale popisuje, kolik z healerova léčení přišlo úplně vniveč. Navíc údaje o léčení/healingu v sobě obsahují i overhealing, takže takto to můžeme oddělit od sebe na efektivní healing a zbytečný healing. To, že má healer veliký overhealing, může znamenat více věcí, například že je příliš dobře vybavený na danou úroveň dungeonu, nebo plytvá své silné schopnosti tehdy, kdy nejsou potřeba s tím, že mu později mohou chybět. Nic z toho však nesevčí o nějakém přímém pochybení.

4.4.4 Léčení v boji/Combat healing

Léčení v boji reprezentuje healing daný během boje s NPC, tudíž tehdy, kdy na něm skutečně záleží. V boji má totiž každý člen týmu jasně přidělenou roli a healing tím pádem primárně spadá jen a pouze na healera. Tento údaj jsme se rozhodli sbírat proto, abychom oddělili léčení v boji od léčení mimo boj, na kterém tolik nezáleží a také není vždy úplně nutné. Spousta postav se totiž může vyléčit i sama, i když nejsou healeři, a proto by healing done mohl být nevyhovující v některých situacích, kdy se nehealeři léčí mimo boj sami, oproti situacím, kdy i léčení mimo boj provádí výhradně healer, například kvůli lenosti hráčů, nebo protože to jejich postava zkrátka neumí.

4.4.5 Přeléčení v boji/Combat overhealing

Přeléčení v boji funguje stejně jako celkové přeléčení, s tou výjimkou, že je počítáno pouze během boje. Tento údaj sbíráme hlavně z důvodu oddělení přeléčení během boje a mimo něj, ať víme, kde k němu primárně dochází, kdy na tom záleží.

4.4.6 Absorbované poškození/Absorbs done

Některé postavy, které spadají do healerské role, mají kromě léčících schopností třeba možnost dát na ostatní hráče štít s omezeným trváním, který je schopen pohltit část poškození, které by jinak bylo přiděleno hráči, na kterém je štít. Jakmile je toto poškození pohlceno, je zaznamenáno absorbované poškození, které je potom přiděleno tomu, kdo vytvořil onen štít.



Obrázek 4.8. Nepřátelský hráč absorbuje poškození štítem [4]

Na obrázku 4.8 můžeme vidět nepřátelského hráče, který absorbuje příchozí poškození svým štítem. Útoky na něj nepůsobí žádný úbytek životů, místo toho mu slábne štít.

4.4.7 Nepřerušená kouzla/Uninterrupted spells

Během dungeonu skupina často narazí na NPC, které čarují nebezpečná kouzla, jejichž čarování je třeba přerušit, protože by mohla skupinu vážně ohrozit. Většina postav spadajících do kategorie damage dealerů nebo tanků má schopnost nebo kouzlo, které spadá do skupiny Interruptů/přerušení. Když hráč použije tuto schopnost nebo kouzlo na NPC, které zrovna něco čaruje, tak tímto čarování přeruší a pošle svůj interrupt na cooldown (nemůže jej po určité době použít znovu). Během dungeonu jsou některá kouzla více nebezpečná než jiná a jejich nepřerušení může mít kritické následky pro skupinu. V addonu je třeba monitorovat, kdy má hráč svůj interrupt připraven k použití. Pokud NPC dokončilo čarování nebezpečného kouzla a hráč měl v této době připravený interrupt, je to přímá chyba, potenciálně vedoucí k neúspěchu skupiny.



Obrázek 4.9. Hráč přerušuje protihráči kouzlení interruptem [4]

Na 4.9 obrázku můžeme vidět protihráče čarovat kouzlo Healing Surge. Hráč použije schopnost kick a čarování kouzla přeruší.

4.4.8 Udělené poškození/Damage Done

Cílem dungeonu je vyčistit jej od určitého počtu trash NPC a všech bossů. Tím pádem je potřeba dát určité množství poškození v co nejkratším čase. Obecně platí, že čím vyšší poškození za sekundu skupina má, tím rychleji dokončí dungeon a také tím pádem bude mít méně prostoru udělat nějakou chybu, a proto damage done v porovnání s časem má přímý vliv na úspěch skupin.

4.4.9 Efektivní poškození/Effective Damage Done

Když tým nezvládne boj s nějakou skupinou nepřátel a všichni postupně v boji umřou, vrátí se skupina nepřátel na své původní místo a všichni se vyléčí na maximální hodnotu svých životů. Všechno poškození utržené v tomto boji tím pádem přichází vniče. Efektivní poškození se započítá hráčům právě tehdy, když zabijí některého z nepřátel a skutečně se potom posunou v dungeonu dál směrem k cíli.

■ 4.4.10 Utržené poškození/Damage taken

DamageTaken reprezentuje poškození, které hráči dostávají v průběhu dungeonu. Různá NPC mívají různé schopnosti, kterým se někdy jde a někdy nejde vyhnout. U těch, kterým se jde vyhnout, se dá u hráčů detekovat jejich chyba. Když hráč dostane poškození, dostává svůj tým pod tlak, protože přidělová práci svému healerovi, který když musí najednou léčit více hráčů což v jednu chvíli nemusí zvládnout a to může vést až ke smrti.

■ 4.5 Technologie addonu

Jako technologii použitou v addonu jsem vybral jazyk Lua, který je prakticky jedinou možnou volbou, protože je v něm naprogramován User Interface hry World of Warcraft, který se kompiluje přímo za běhu hry v ní.

Lua je mocný, efektivní, lightweight, embeddable skriptovací jazyk. Podporuje procedurální programování, objektově orientované programování, funkcionální programování, data-driven programování a data description. Lua kombinuje jednoduchý procedurální syntax se silnými data description konstrukty založenými na asociativních polích a rozšiřitelných semantikách. Lua je dynamicky typovaná, běží interpretací bytekódu s register-based virtual machine a má automatickou správu paměti s inkrementální garbage collection, což ji dělá ideální pro konfiguraci, skripting a rapidní prototyping. [19]

Lua je používána ve spoustě komerčních aplikacích (např. Adobe's Photoshop Lighroom), s důrazem na embedded systémy (např. the Ginga middleware for digital TV in Brazil) a hry (např. World of Warcraft and Angry Birds). Lua je momentálně hlavním skriptovacím jazykem v hrách. Lua má solidní referenční manuál a je o ní napsaných několik knížek. Několik verzí jazyka bylo vydáno a použito v reálných aplikacích od doby jejího vytvoření v roce 1993. Lua nechyběla na HOPL III, the Third ACM SIGPLAN History of Programming Languages Conference, in 2007 a vyhrála Front Line Award 2011 od Game Developers Magazine. [19] Ve hře World of Warcraft se používá ke tvorbě uživatelského rozhraní. Lua skripty se dají spouštět přímo ve hře skrze herní chat, nebo se předem napsané skripty dají spustit jako addon (uživatelské custom rozšíření UI ve hře, jakýsi uživatelem vyrobené plugin) přímo přidáním skriptu do adresářové struktury hry. Skrze Lua skripty se dají ve hře zachytávat herní eventy a k nim se dají přiřazovat vlastnoručně napsané event handling funkce. Takto můžeme získat přístup k logům hry v reálném čase, ze kterých lze zjistit co konkrétně který hráč v okolí provádí a co se mu stalo/děje. Užití jazyka Lua ve hře World of Warcraft spočívá hlavně na event driven programování. Programátor se přihlašuje k odběru různých eventů generovaných hrou a ty potom spouští logiku a běh addonu.

Na obrázku 4.10 můžeme vidět error při interpretování kódu addonu ve hře World of Warcraft. Tímto způsobem se zobrazují chybové hlášky programátorům pro debugování kódu.



Obrázek 4.10. Lua error ve hře World of Warcraft [4]

4.6 Technologie frontendu

4.6.1 Angular

Angular [20] byl poprvé vydán v roce 2010 Googlem, tehdy ještě pod názvem AngularJS. Dnes se jedná o javascriptový framework využívající jazyk TypeScript, který se transpiluje do javascriptu. Jakýkoliv validní Javascript je přitom validní Typescript. Typescript je supersetem Javascriptu. Angular je vývojová platforma postavená na TypeScriptu. Jako platforma v sobě Angular zahrnuje:

- Framework postavený na komponentách pro stavění škálovatelných webových aplikací
- Kolekci dobře integrovaných knihoven, které pokrývají širokou škálu vlastností, zahrnujících routing, manipulaci s formuláři, klient-serverovou komunikaci a další
- Soubor vývojářských nástrojů pro pomoc s vývojem, vytvářením, testováním a aktualizací kódu

S Angularem využíváte platformy, která se dokáže škálovat od projektů pro jednoho vývojáře až na enterprise-level aplikace. Angular je navržen tak, aby aktualizace kódu probíhaly co nejlépeji to jde, aby jste mohli využít novinek s minimálním úsilím. K tomu všemu se Angular ekosystém skládá z různorodé skupiny developerů, autorů knihoven a content creatorů, kterých je dohromady přes 1,7 milionu. [20] Angular je dnes například používán na těchto projektech a těmito společnostmi: Webové aplikace Microsoft Office, Deutsche Bank, Mixer, Gmail, Forbes, Paypal [21]

4.6.2 React

React [22] byl poprvé vydán v roce 2013 Facebookem, který jej hojně využívá napříč svými aplikacemi. Jedná se o javascriptovou knihovnu pro vytváření uživatelských rozhraní.

React má obrovskou uživatelskou základnu (viz srovnání dále).

Některé společnosti využívající React:

- Netflix
- Dropbox
- Airbnb
- Reddit
- BBC

■ 4.6.3 Vue

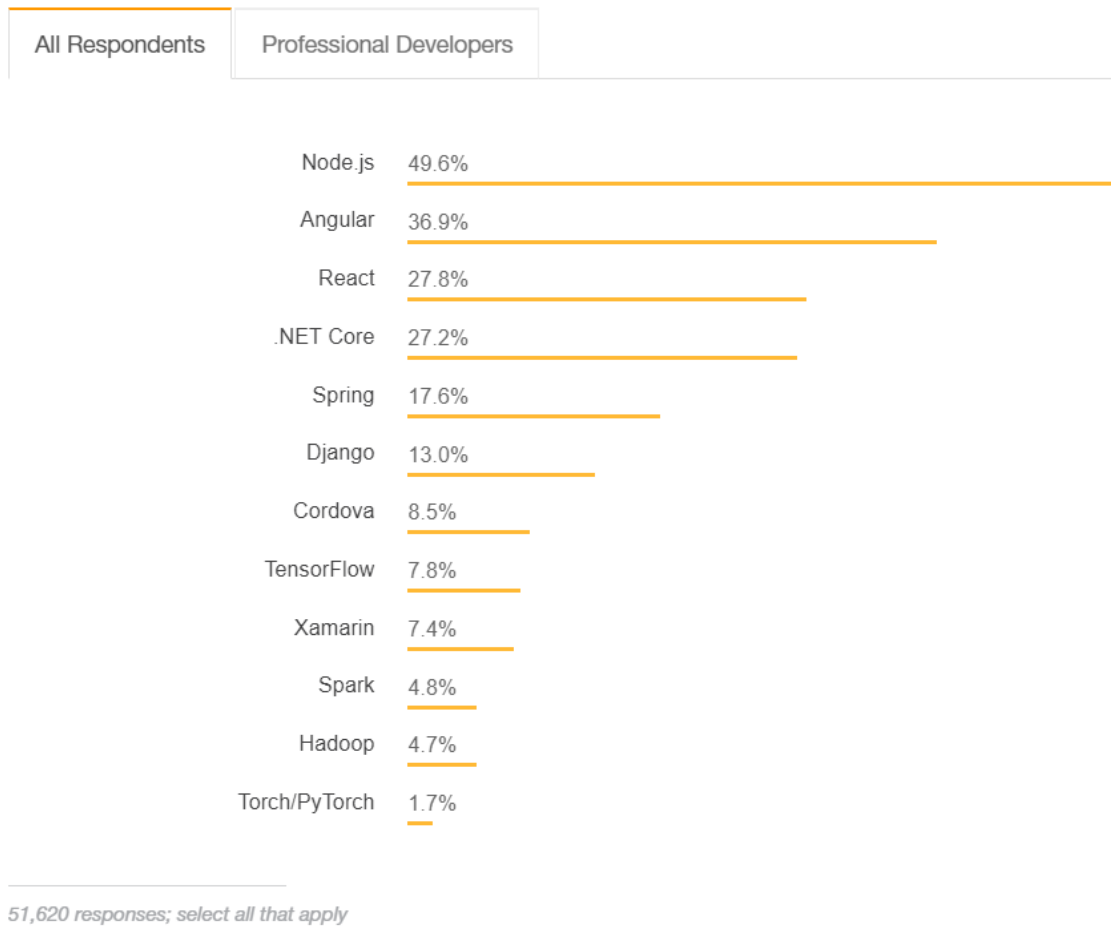
Vue [23], také známo jako Vue.js je nejmladším členem trojice Angular, React, Vue. Bylo vyvinuto bývalým zaměstnancem Googlu Evanem You v roce 2014. Za poslední tři roky zažilo Vue obrovský nárůst popularity, a to i přes to, že nemá základ a podporu velké společnosti. Momentální stabilní verze je 3.0, vydaná v září 2020 (s několika malými inkrementálními releasy od té doby). Vývojáři Vue jsou podporováni přes Patreon. Mělo by být zmíněno, že Vue 3 se momentálně nachází ve vlastním GitHub repozitáři a přešlo na jazyk TypeScript. [24]

I přes to, že je Vue takto mladé, našlo si velké zastoupení u velkých společností v komerční sféře. Mezi uživatele Vue dnes například patří tyto společnosti:

- Google - jejich stránka careers.google.com je napsána ve Vue
- Apple - použil Vue na jejich SwiftUI tutorial developer.apple.com/tutorials/swiftui
- GitLab - celý GitLab je doručeny jako jedna aplikace ve Vue
- Zoom - Zoom používá Vue pro jejich webovou aplikaci [25]

4.6.4 Porovnání

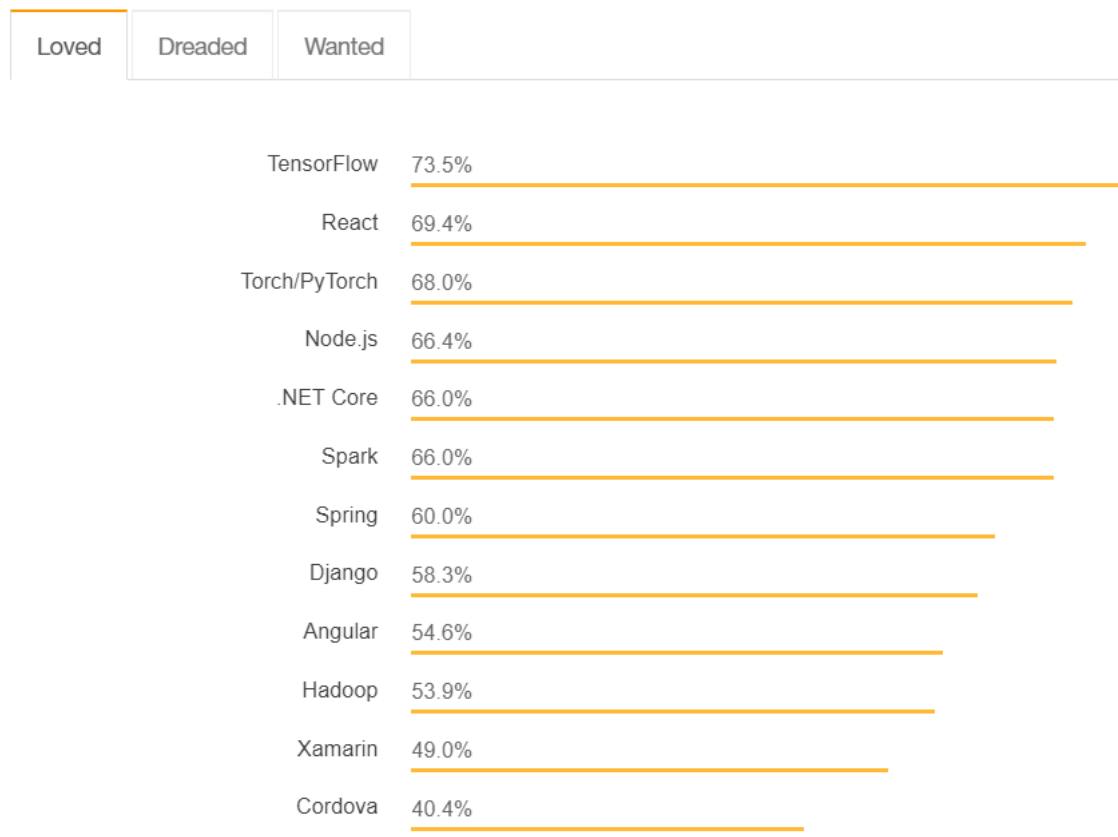
Frameworks, Libraries, and Tools



Obrázek 4.11. Statistika používanosti frameworků [26]

Podle statistiky 4.11 na StackOverflow je Angular používanějším frameworkem, než React, kdežto Vue se na top příčky ani nedostalo.

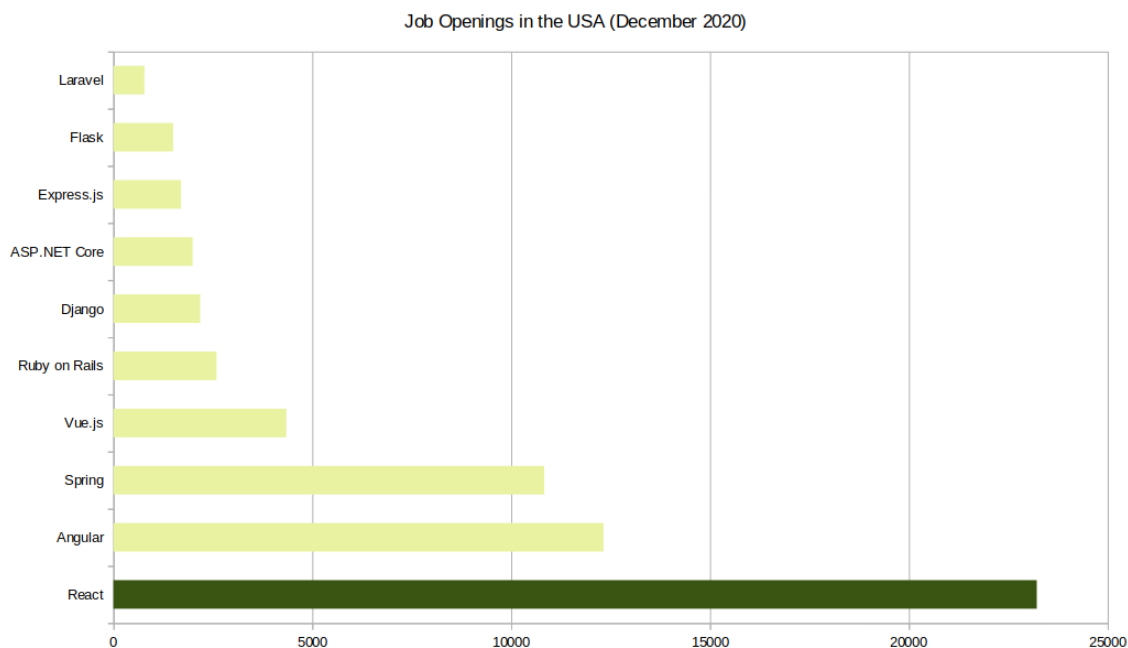
Most Loved, Dreaded, and Wanted Frameworks, Libraries, and Tools



% of developers who are developing with the language or technology and have expressed interest in continuing to develop with it

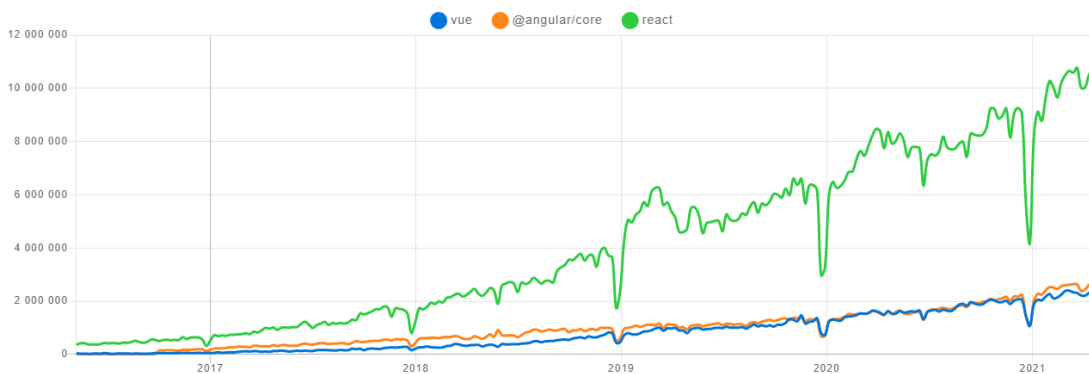
Obrázek 4.12. Statistika oblíbenosti frameworků mezi developery [26]

Podle statistiky 4.12 na StackOverflow je React “milovanějším frameworkem”, to znamená, že lidé současně pracující s tímto frameworkem projevili zájem o pokračování v práci s tímto frameworkem. Vue je sice na vzestupu, ale pořád potřebuje ještě trošku času aby si našlo větší základnu uživatelů.



Obrázek 4.13. Příležitosti práce v různých frameworkích v USA [27]

Dle statistik 4.13 jsou oba frameworky, Angular i React velice žádané na trhu práce v USA, z čehož se dá určitě vyvodit, že budou populární i ve zbytku světa. Nicméně React poráží Angular i v tomto aspektu. Vue tady nezaostává o tolik a ukazuje se, že je čím dál více žádanější. Statistika ukazuje, že všechny tři frameworky jsou hojně používané a i nejméně používané Vue je jeden z nejpoužívanějších frameworků dneška.



Obrázek 4.14. Stahovanost 3 js frameworků z NPM [28]

Stahovanost frameworků z NPM (node package manager, sloužící pro distribuci programátorských knihoven, modelů a frameworků), zobrazená na obrázku 4.14, také jasně dokazuje, že React je dnes používanější z frameworků. Vue je na tom nicméně podobně jako Angular.

Na tabulce 4.2 vidíme statistiky jednotlivých frameworků na GitHubu.

4.6.5 Závěr

Pro technologii použitou na frontendu jsem se rozhodl pro Angular, hlavně proto, že s ním mám již zkušenosti z minula a vím, že přesně dostojí požadavkům na frontend systému. Mimo jiné je Angular v praxi velmi používaný a patří k momentálním trendům

	Angular	React	Vue
Watchers	3.2k	6.7k	6.3k
Stars	70.9k	164k	200.8k
Forks	18.6k	32.9k	31.7k
Contributors	1352	1533	382

Tabulka 4.2. Statistiky GitHub repozitářů těchto frameworků [24].

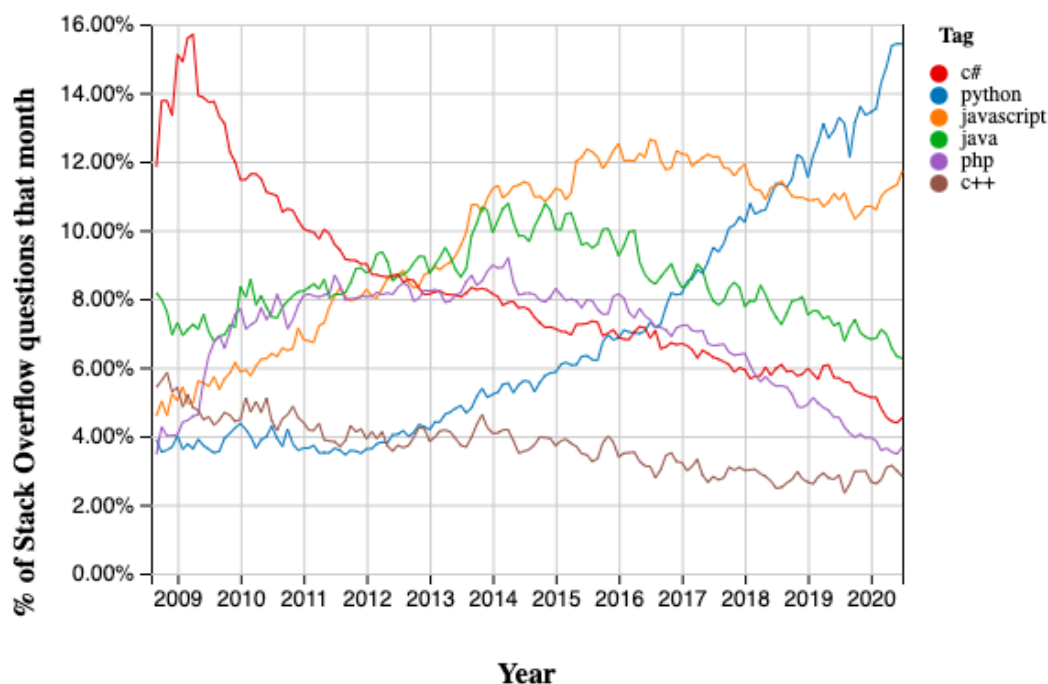
ve frontendových technologiích. Toto tvrzení je dále dokázáno, že jej používají společnosti jako Paypal, Forbes, Overleaf, Samsung nebo Gmail. Ačkoliv oproti Reactu a Vue je těžkopádnější, já osobně v něm budu mnohem obratnější a kód budu produkovat rychleji.

4.7 Technologie predikce

4.7.1 Python

Python je dnes jedním z nejoblíbenějších programovacích jazyků vůbec a má velkou programátorskou základnu. Pro strojové učení je nejpreferovanějším. [7] Je přátelský k začátečníkům, snadněji pochopitelný a jeho kód je čtivý oproti jiným jazykům. Python také nabízí širokou škálu různých knihoven pro manipulaci s daty a strojové učení nebo různé vykreslovací a data kreslící knihovny. Mezi tyto knihovny například patří:

- Numpy - poskytuje infrastrukturu pro práci s vektory, manipulace s daty [29]
- Theano - manipulace s matematickými výrazy, nástavba na Numpy
- Scikit-learn - knihovna obsahující implementaci klasifikačních algoritmů [30]
- PyTorch - knihovna podporující implementaci neuronových sítí [31]



Obrázek 4.15. Zastoupení programovacích jazyků v otázkách na stack overflow [32]

Na obrázku můžeme vidět, že popularita pythonu na Stack Overflow stoupá a je to jeden z nejpobulárnějších jazyků dneška.

■ 4.7.2 Java

Java je dnes také široce používaná a to nejen na poli strojového učení. Dle obrázku výše je třetím nejdotaovanějším jazykem na Stack Overflow. Ohledně strojového učení je druhým nejpreferovanějším jazykem. [33] Ačkoliv co se produktivity zaostává za Pythonem, pořád jde o poměrně čitelný jazyk. Java také nabízí širokou škálu knihoven a nástrojů určených pro použití ve strojovém učení. Mezi ty například patří:

- Deep learning for Java [34]
- Apache spark's MLlib [35]

■ 4.7.3 Závěr

Pro technologii pro predikci jsem se rozhodl pro Python, protože je to nejpobulárnější jazyk pro strojové učení a má největší komunitu a tím pádem i online podporu. S pythonem a numpy už mám také zkušenosti a bude to pro mě jednodušší než začínat s Javou ve strojovém učení od začátku. Taky mám pro některé algoritmy už kódy předpřipravené. Zároveň jsme se rozhodli pro framework Django na zbytku backendu, což je framework postavený na jazyce Python což nám umožňuje mít byznys logiku jako jeden celek na serveru.

■ 4.8 Databáze

Další technologií, kterou bylo třeba zanalyzovat byl typ databáze. Z obrázku 4.5 můžeme vidět strukturu dat, která se bude vyskytovat v databázi. Jeden aktivní hráč World of Warcraftu odejde dle Martinova pozorování za týden kolem 10 běhů módu Mythic+. V únoru 2021 bylo odhadem zaukrouhleně na statisíce 5 200 000 aktivních hráčů. Za předpokladu, že by náš doplněk nabyt popularity, se rozhodně jedná o big data a bylo by na místě zvolit NoSQL typ databáze. Jelikož ale Django přímo podporuje jen SQL databáze a hráčů bude ze začátku systém využívat málo, nechali jsme tento předpoklad do budoucna a zvolili jsme databázi SQL. [10]

■ 4.9 Verzování

Na serverové části aplikace se nachází byznys logika (logika predikce) a zároveň manipule s databází, zpracovávání dat nasbíraných addonem do hry a také obsluha requestů frontendu. Na systému spolupracujeme ve dvojici a právě na serverové části systému oba přispíváme do stejného kódu. Z tohoto důvodu vznikla potřeba verzování této části KeystonePro systému.

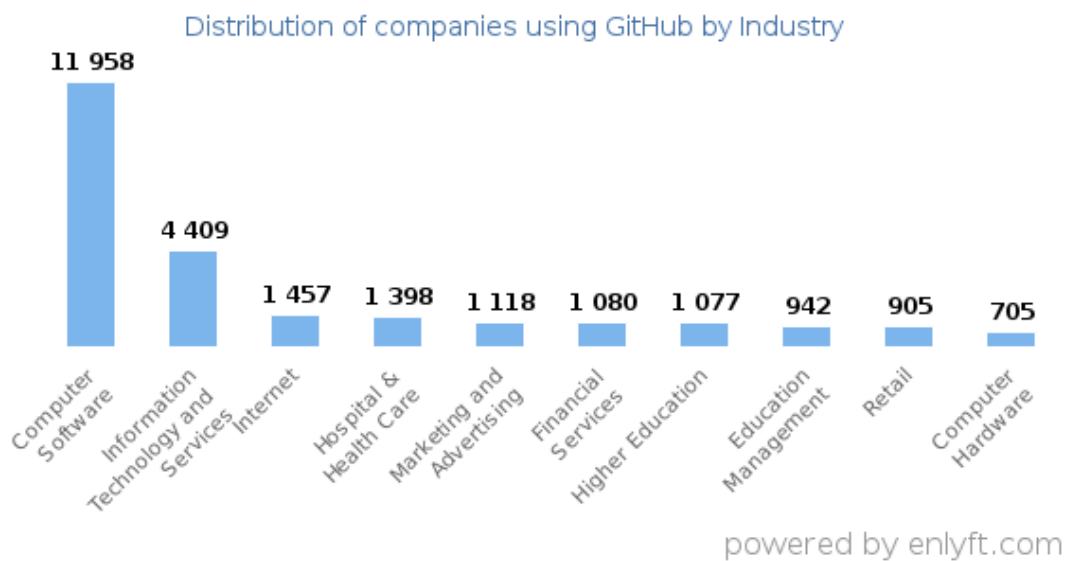
■ 4.9.1 Git

Git je bezplatný a open zdrojový distribuovaný version control systém navržený ke zvládnutí všeho od malých po obrovské projekty s rychlostí a efektivitou. Git je jednoduchý na naučení a má velice malý footprint s bleskově rychlým výkonem. Převyšuje SCM nástroje jako Subversion, CVS, Perforce a ClearCase s nástroji jako cheap local branching, pohodlné staging areas a více možnostmi workflow. [36]

Git se dá lehce naučit z knížky Pro Git, která slouží také jako dokumentace. Je volně dostupná na webu Gitu.

Git je jedinečný například ve svém branchingovém systému, kde si programátor může lokálně založit mnoho různých větví projektu, které jsou na sobě nezávislé.

Existuje mnoho git hostingových stránek, jako například Gitlab, Bitbucket od Atlassian nebo Github, který je zároveň místo, kde se schází obrovská komunita vývojářů. Git je dnes tak rozšířený, že při zakládání projektu v některých frameworkcích se automaticky vytvoří i gitový repozitář. Spousta moderních IDE a programovacích editorů, jako VSCode nebo IntelliJ Idea mají zakomponované rozhraní pro práci s gitem přímo v sobě. Git je široce používán mezi amatéry, ale i mezi komerčními společnostmi nejen v IT oboru.



Obrázek 4.16. Rozdělení zákazníků GitHubu na odvětví jejich byznysu [37]

Při pohledu na zákazníky GitHubu podle odvětví jejich byznysu zobrazené na obrázku 4.16, nacházíme, že počítačový software (22%) a informační technologie a služby (8%) jsou největšími segmenty.

■ 4.9.2 Závěr

Pro verzování jsme použili git s hostingem přímo Heroku. A to hlavně z důvodu, že na Heroku deployujeme náš backend i frontend. Hosting gitu na Heroku je tím pádem pro nás přirozenou a nejjednodušší volbou.

Kapitola 5

Dataset

Jeden záznam v datasetu se rovná jednomu dokončenému dungeonu.

Data v datasetu mohou podléhat biasu. Já i můj kolega Martin se ve hře pohybujeme v uzavřených komunitách nám blízkých a naši známí jsou již zkušenější hráči, kteří nejsou ve hře vůbec noví. Hráči z těchto komunit nám pomáhali se sběrem dat do datasetu. Protože se jedná o zkušené hráče, očekáváme, že dataset bude obsahovat menší zastoupení klíčů malých úrovní, kterých bude ve skutečném světě podstatně více. Dále čísla v datasetu budou pravděpodobně vypovídat o lepších a kvalitnějších výkonech hráčů, než bude reálný průměr.

Celkem jsme nasbírali 74 záznamů dungeonů v poměru 50:24, 50 úspěchů a 24 neúspěchů neboli depletů. V procentech to je 67.56% ku 32.44% zaokrouhleno na setiny procenta. Nasbíraná data jsou nevyvážená s převahou třídy úspěchů. Toto je zapříčiněno například tím, že když je během dungeonu jasné, že se nestihne včas, protože hráči v týmu neustále chybují, někteří hráči opustí skupinu a vzdají dungeon v průběhu. Protože nedošlo k dokončení, tak nejsou posbíraná žádná data. Také je to dané tím, že hráči ve skupině, se kterou jsme sbírali data, se primárně věnují jinému hernímu módu a mód Mythic+ volí spíše jako oddech a dávají tak často přednost pro ně jednodušším klíčům s nižší úrovní (cca 14), než těžším klíčům s vyšší úrovní (cca 16-18), protože od úrovně 14 už se odměny za dokončení klíče neliší.

Abych zlepšil přesnost predikce na minoritní třídě, rozhodl jsem se náhodně odstranit 10 vzorků dat z majoritní třídy. Nový poměr dat v datasetu dává tedy 62.5% ku 37.5%.

Data v datasetu mají následující sloupce/featuresy, které se spočítají z nasbíraných dat, uvedených v kapitole 4.

5.1 InterruptScore

- Průměr: 48.781250
- Standartní odchylka: 46.393578
- Minimum: 0
- Maximum: 255
- 25 percentil: 20
- Medián: 35
- 75 percentil: 49

Kdykoliv nepřátelské NPC v dungeonu vyčaruje kouzlo, které je na seznamu nebezpečných kouzel a hráč měl možnost toto čarování přerušit svým přerušením/interruptem, započítá se mu missedInterrupt. InterruptScore je potom součet missedInterrupt u všech hráčů v týmu. Atributu nabývá kladných, celočíselných, hodnot zdola omezených nulou.

5.2 DamageDoneScore

- Průměr: 28698880
- Standartní odchylka: 8750108
- Minimum: 13486940
- Maximum: 52136680
- 25 percentil: 21850330
- Medián: 27329870
- 75 percentil: 34772200

Jedná se o součet EffectiveDamageDone hráčů vydělený completionTimeRatio a zaokrouhlený na celá čísla. Účel tohoto atributu je charakterizovat rychlost postupu týmu. Jedná se o celočíselný, kladný atribut zdola omezený nulou.

5.3 DeathsScore

- Průměr: 12.625
- Standartní odchylka: 13.369594
- Minimum: 0
- Maximum: 80
- 25 percentil: 3.75
- Medián: 8
- 75 percentil: 18.25

Pokaždé, když hráč umře, započítá se mu jedna smrt. Ve skupině se jedná o součet těchto smrtí. Jedná se o kladný, celočíselný atribut zdola omezený nulou.

5.4 DamageReceivedScore

- Průměr: 11757610
- Standartní odchylka: 2962380
- Minimum: 5797060
- Maximum: 21602620
- 25 percentil: 9858500
- Medián: 12028520
- 75 percentil: 13085430

Jedná se o atribut spočítaný dělením totalDamageReceived daného hráče completionTimeRatiem a poté se výsledek zaokrouhlí na celá čísla. Nakonec se pro tým sečtou hodnoty u všech hráčů. Jedná se o kladný, celočíselný atribut zdola omezený nulou.

5.5 FailureDamageScore

- Průměr: 1168682
- Standartní odchylka: 497871.2
- Minimum: 352019
- Maximum: 2775399
- 25 percentil: 881780

- Medián: 1084662
- 75 percentil: 1389479

Nepřátelé v módu Mythic+ používají spoustu kouzel, avšak ne všechna jsou nebezpečná, nebo se jim dá nějakým stylem vyhnout. Pokud je však kouzlo nebezpečné a dá se mu vyhnout, přidal jsem ho do tabulky tzv. `failure_spells`, dále jsem ještě vytvořil tabulku `failure_spells_no_tank`, kde se vyskytují vyhnutelná nebezpečná kouzla, která ale ohrozí hráče jedině pokud nemají roli TANK. Prošel jsem všechny záznamy o obdržení poškození a podíval jsem se, zda je jejich zdroj stejný jako v tabulce, pokud ano, poškození jsem přičetl do složky tohoto vektoru. [10] Jedná se o součet těchto hodnot od všech hráčů.

5.6 FailureDamageRatio

- Průměr: 0.160839
- Standartní odchylka: 0.190621
- Minimum: 0.026467
- Maximum: 0.953319
- 25 percentil: 0.081990
- Medián: 0.094996
- 75 percentil: 0.124634

FailureDamageRatio je pouze poměr složek `failureDamageScore` a `damageReceivedScore` vyjádřený jako desetinné číslo. [10] Jedná se o spojitý atribut zdola omezený nulou a shora jedničkou.

5.7 HealingScore

- Průměr: 10792020
- Standartní odchylka: 2872439
- Minimum: 5482263
- Maximum: 20981760
- 25 percentil: 8687287
- Medián: 10892160
- 75 percentil: 12544920

Tento atribut je spočítán součtem `totalHealingDone` s `totalAbsorbsDone`, od kterého je poté odečteno `totalOverhealingDone`. Celý tento výraz je pak vydělen složkou `completionTimeRatio` a zaokrouhlen na celá čísla. [10] Nakonec se pro tým sečtou hodnoty u všech hráčů. Jedná se o kladný, celočíselný atribut zdola omezený nulou.

5.8 DungeonLevel

- Průměr: 12.625
- Standartní odchylka: 3.543382
- Minimum: 5
- Maximum: 24
- 25 percentil: 11
- Medián: 13

- 75 percentil: 14

Jedná se o úroveň daného dungeonu. Čím vyšší úroveň, tím vyšší jsou nároky na jeho dokončení a tím těžší je jej dokončit včas. Jedná se o diskrétní, celočíselnou množinu omezenou zdola dvojkou (nejnižší úroveň klíče ve hře je 2).

5.9 DungeonName

Na tomto seznamu vidíme četnost dungeonů daného typu v našem datasetu.

- Theater of Pain: 15
- Halls of Atonement: 6
- Plaguefall: 6
- Mists of Tirna Scithe: 6
- Spires of Ascension: 10
- Sanguine Depths: 5
- The Necrotic Wake: 12
- De Other Side: 4

DungeonName nám říká typ dungeonu. Každý dungeon je trochu jiný a vyžaduje od hráčů trochu jiný skillset. Typy dungeonů se také liší časy na jejich dokončení, nebo počtem bossů. NPC v každém dungeonu mají jiné mechaniky, na které je třeba dávat pozor. Hráči se musí individuálně naučit každý z těchto dungeonů a naučení se mechanik v jednom dungeonu nedává hráči žádnou výhodu v dungeonu jiném. Hodnota atributu DungeonName může nabývat osmi různých hodnot, protože je osm různých typů dungeonu - De Other Side, Halls of Atonement, Mists of Tirna Scithe, The Necrotic Wake, Plaguefall, Sanguine Depths, Spires of Ascension a Theater of Pain.

5.10 CompletionTimeRatio

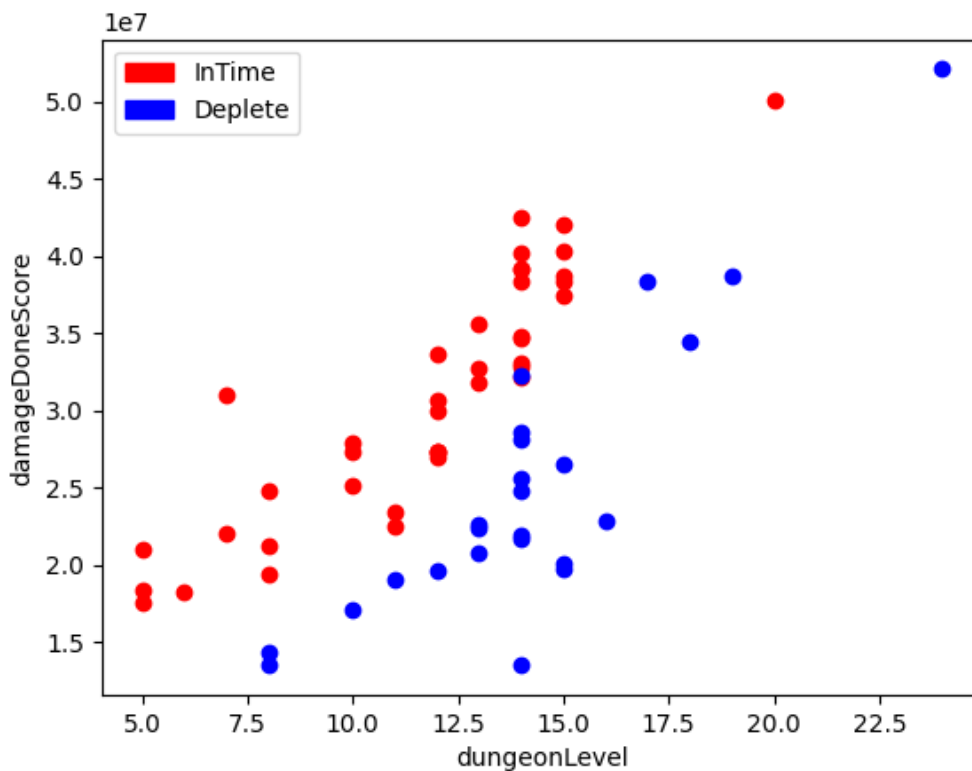
- Průměr: 0.953120
- Standardní odchylka: 0.297608
- Minimum: 0.585164
- Maximum: 2.290705
- 25 percentil: 0.755012
- Medián: 0.867287
- 75 percentil: 1.070157

Každý dungeon má jiný časový limit, proto je hodnota z nasbíraných dat o dokončení dungeonu v milisekundách sama o sobě nic neříkající. Aby se dala data porovnávat všechna mezi sebou, musí být v poměru vůči časovému limitu konkrétního dungeonu. [10] Jedná se o kladný nenulový spojitý atribut.

5.11 Vizualizace dat

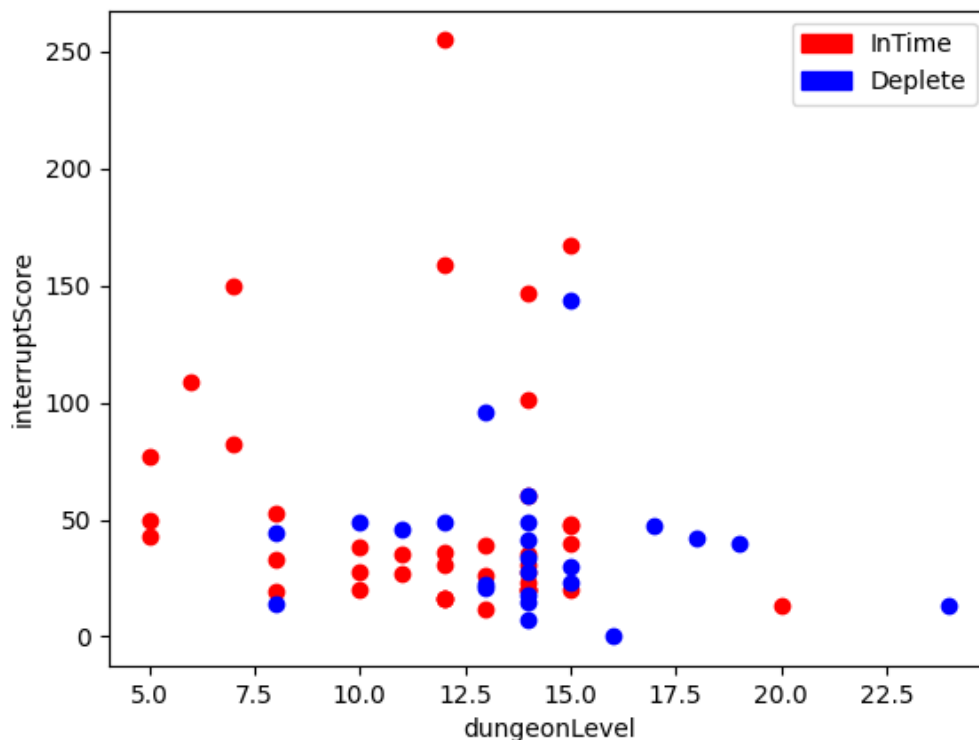
Data jsem jsem zobrazoval pomocí PCA (Principal component analysis) a zkoušel různé kombinace sloupců. Dataset se nejlépe zobrazil použitím sloupců Dungeon Level, FailureDamageRatio a DamageDoneScore (zobrazení pomocí těchto tří sloupců vidíme na obrázku 5.6), což dává smysl, protože úspěch v zásadě závisí na tom, kolik tým udělá

chyb, což je zachyceno ve FailureDamageRatio a potom samozřejmě čím víc dává damage, tím rychleji eliminuje nepřátelské NPC a rychleji postupuje dungeonem a je to právě rychlost postupu, která závisí na tom, jestli se dungeon zvládne včas. Na dungeon level závisí, protože čím vyšší úroveň, tím víc potřebuje tým dát damage a tím méně si může dovolit udělat chyb. Co může stačit na úroveň 10 zároveň nemusí stačit na úroveň 15. To můžeme pozorovat na grafu 5.1 níže. Jedná se o graf DamageDoneScore v závislosti na DungeonLevel. Červeně jsou zobrazeny dungeons dokončené včas, modře jsou zobrazeny dungeons nedokončené včas. Můžeme pozorovat, že nároky na DamageDoneScore s úrovní dungeonu rostou.



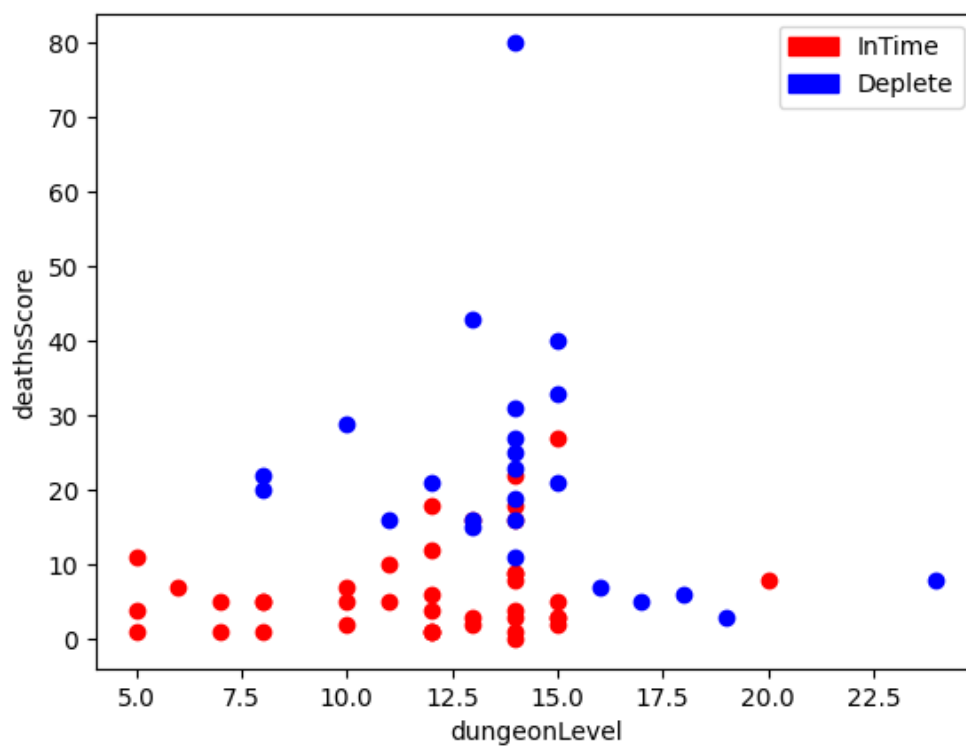
Obrázek 5.1. Závislost DamageDoneScore na DungeonLevel.

InterruptScore necharakterizuje data tak dobře, jak jsme čekali, a to z hlavně z důvodu, že přerušování/interruptování čarování NPC závisí hodně na typu dungeonu. V některých dungeonech jsou více nebezpečná kouzla, v jiných je jich o mnoho více, ale jsou méně nebezpečná a v některých dungeonech není třeba přerušovat téměř nic, kdežto v jiných to je klíčové a každý netrefený interrupt může znamenat smrt hráče. Každé kouzlo také dělá trochu něco jiného a když je tým připraven na následky, tak nepřerušení nemusí být zas takový problém. To, že InterruptScore nepopisuje dobře nároky na zvládnutí dané úrovně dungeonu můžeme vidět na následujícím grafu 5.2.

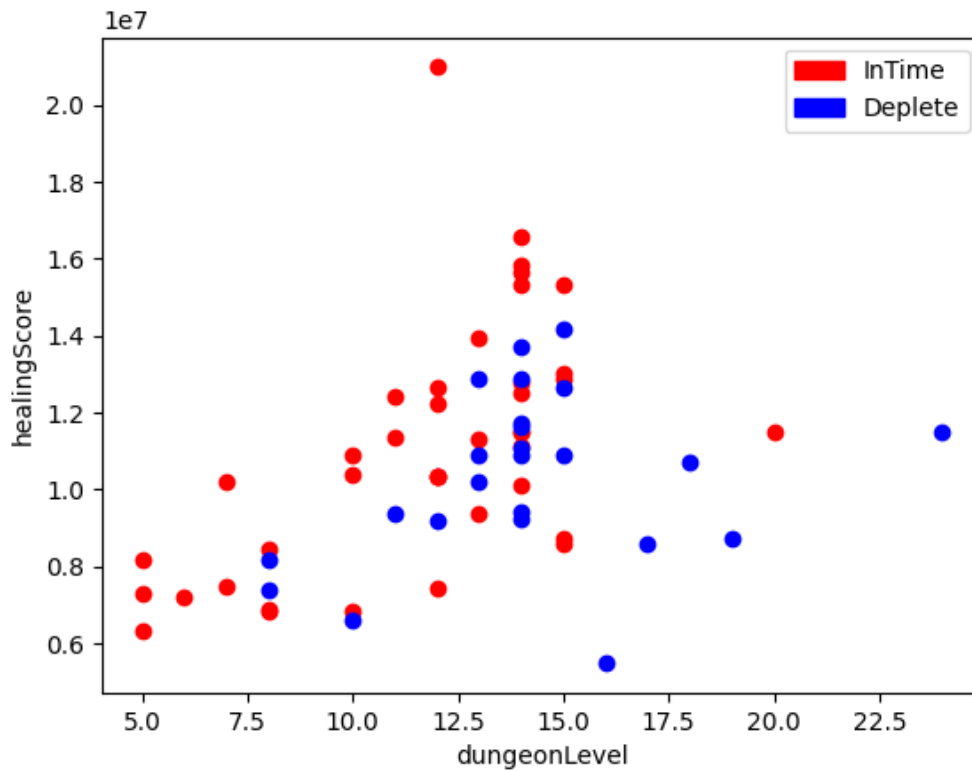


Obrázek 5.2. Závislost InterruptScore na DungeonLevel.

DeathsScore také nerozděluje data tak dobře, jak se nám původně zdálo, a to může být kvůli toho, že smrt hráče hned na místě, kde se může oživit není takový problém, jako když musí běžet i několik minut, aby se vrátil zpátky do boje, nebo když umře na začátku boje s bossem, poté musí tým bosse zdolávat ve 4 a to opravdu obtížné a to i na tolik, že se mnohdy vyplatí zkrátka nechat všechny co nejrychleji umřít a začít novým pokusem. Také záleží, jestli hráči umírají postupně po jednom, nebo všichni naráz. Když umře celý tým najednou, dojde k resetu nepřátelských NPC a všechny se vrátí na svou původní pozici a vyléčí si životy do plna. Přes tyto nevýhody pořád můžeme na grafu 5.3 níže vidět, že až na výjimky se počet tolerovaných smrtí pro úspěch snižuje s úrovní dungeonu. To, že když hráč hodně umírá, tak nedává veliký damage, nemusí nutně platit, protože damage záleží hodně na vybavení hráče, nebo jak jsem zmínil výše, jak dlouho se po smrti vrací zpátky do boje - jak moc ho smrt zpomalila. To můžeme pozorovat na grafu závislosti DamageDoneScore na DeathsScore 5.4, kde vidíme, že i když hráči umřou 20x - 30x během dungeonu (což je opravdu hodně, průměr DeathsScore je 12.625), pořád jsou schopni rozdat rozumný damage a klíč tak dokončit včas.

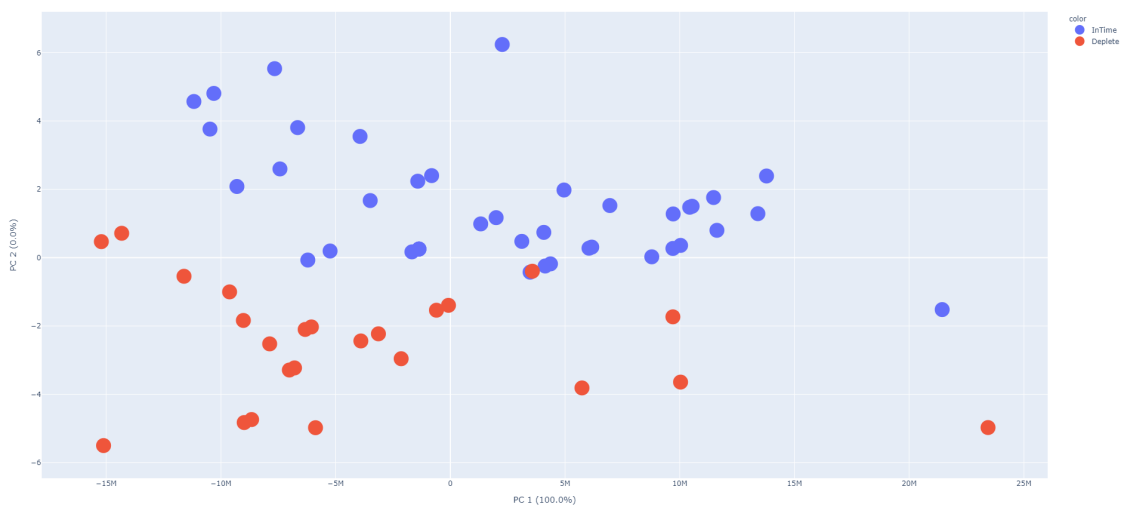


Obrázek 5.3. Závislost DeathsScore na DungeonLevel.



Obrázek 5.5. Závislost HealingScore na DungeonLevel.

Na obrázku 5.6 níže můžeme vidět znázornění datasetu pomocí metody PCA dle sloupců DamageDoneScore, FailureDamageRatio a DungeonLevel. Na grafu se úspěchy a neúspěchy rozumně oddělují do dvou oblastí a jde na něm také rozeznat nárůst nároků s úrovní klíče, kdy na pravé části grafu se vyskytují spíše vyšší úrovně a na levé zase spíše ty nižší.



Obrázek 5.6. Zobrazení dat podle PCA na základě sloupců DamageDoneScore, FailureDamageRatio a DungeonLevel

5.12 Posudek vzorků dat

Pro posudek vzorků dat jsem si vybral náhodně jeden vzorek dat z třídy úspěchů a jeden vzorek z třídy neúspěchů/depletů.

5.12.1 Sanguine Depths 14 - InTime

- interruptScore : 31
- damageDoneScore : 42481189
- deathsScore : 9
- damageReceivedScore : 16460126
- failureDamageScore : 1191972
- failureDamageRatio : 0.07241572755882913
- healingScore : 15670419
- dungeonLevel : 14
- dungeonName : Sanguine Depths
- completionTimeRatio : 0.7416016260162601

Interrupt skóre v tomto konkrétním dungeonu je 31, což je lehce pod mediánem, který je 35. V porovnání s ostatními dungeony je to ničím nevyčívající výsledek. Interrupt skóre je hodně závislé na typu dungeonu, proto jsem ho porovnal s ostatními znanými Sanguine Depths. Z tohoto porovnání vychází tento dungeon run z daleka nejlépe - ostatní Interrupt skóre v dungeonech typu Sanguine Depths jsou 60, 101, 159 a 167 a všechny jsou tak nad hranicí 75 percentilu.

DamageDoneScore je v tomto případě jedno z nejvyšších. 75 percentil je v této kategorii 34772200 a tento vzorek je ještě poměrně vysoko nad touto hodnotou. Z tohoto údaje můžeme vyvodit, že skupina postupovala dungeonem opravdu rychle.

DeathsScore 9 je sice menší než je průměr, nicméně průměr deathscore hodně zvedají extrémně nepovedené dungeon runy. Medián DeathsScore je 8, takže se nejedná o nic zázračného. DamageDoneScore skupiny je pořád opravdu vysoké, což znamená že smrti byly dost pravděpodobně na “správném místě” a ve “vhodný čas”, takže to postup skupiny příliš nezpomalilo.

DamageReceivedScore je v tomto případě opravdu vysoké. Tato skupina musela snášet během dungeonu vysoký damage. Hodnota 16460126 se nachází vysoko nad 75 percentilem. Utržené poškození se dá však zvládat, pokud jej skupina dostává postupně a healer skupiny je na to připravený. Do DamageReceivedScore se počítá i damage do tanka a ten se poškození sice snaží minimalizovat, ale jeho role spočívá v tom, že dokáže právě obrovské množství ustát.

FailureDamageScore je v tomto případě lehce nadprůměrné a také vyšší, než je medián. Nicméně jedná se o dungeon 14 úrovně a když v tomto dungeonu uděláte chybu, je už to dost poznat a velké chyby vás většinou zabijí okamžitě. V nižších úrovních vám chyba jednoduše nedá tolik poškození, a proto i když uděláte více chyb, může tenhle atribut vyjít menší. Průměr FailureDamageScore mezi všemi dungeony úrovně 14 v tomto datasetu je 1221251 a medián 1137533. Pořád se sice tento dung v tomto atributu nachází nad mediánem, ale průměr i 75 percentil o hodnotě 1363384 už je vyšší, takže se jedná o neutrální výsledek.

FailureDamageRatio je v tomto případě hodně malinké, což svědčí o tom, že hráči nedělali mnoho chyb. Tento výsledek se nachází v prvním kvartilu tohoto atributu, 25 percentil je 0.081990. Z tohoto můžeme vyvodit, že se jednalo o opravdu dobře a čistě zahrany dungeon.

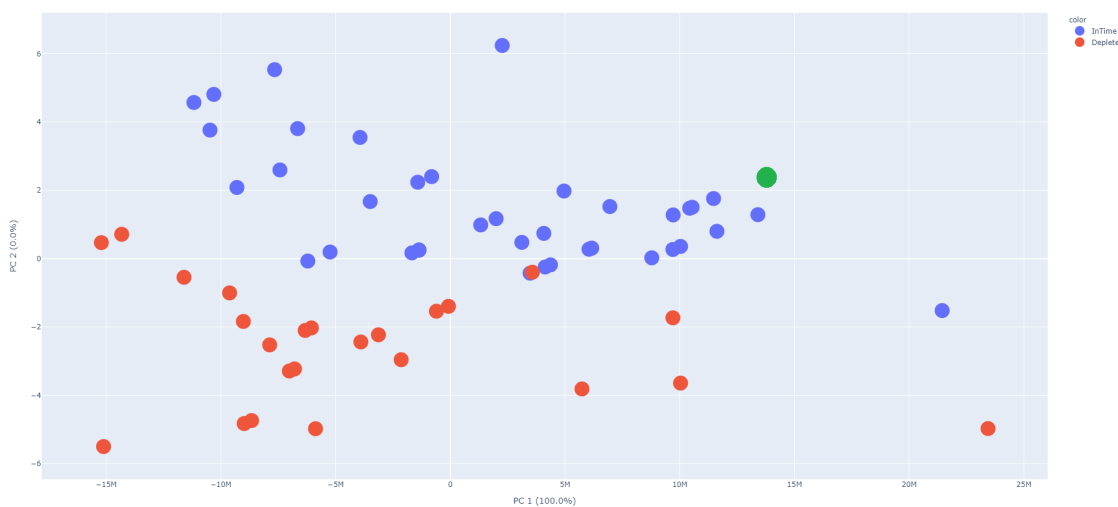
HealingScore 15670419 je v tomto případě opravdu vysoké a nachází se nad 75 percentilem, který má hodnotu 12544920. Tohle může znamenat více věcí, ale v každém případě healer této skupiny je kvalitní hráč.

DungeonLevel tohoto dungeon runu je 14. Úrovně 14 už nejsou jednoduché, bývá to hranice průměrných hráčů. Chyby na této úrovni jdou poznat a nedává-li hráč pozor, může snadno umřít. Aby hráč zdolal dungeon úrovně 14, měl by znát veškeré mechaniky, které se v něm vyskytují. Může si ještě dovést dělat pravidelně chyby malé, ale nesmí jich udělat více v rychlém sledu po sobě. Hrubé chyby už jsou na úrovni 14 trestané okamžitou smrtí hráče. V našem datasetu máme 18 záznamů úrovně 14, z toho 11 je včas, což odpovídá 61% úspěšnosti, zaokrouhlo na celá procenta.

CompletionTimeRatio tohoto dungeonu je 0.7416016260162601. 25 percentil je 0.767556, takže tenhle záznam patří mezi 25% nejrychlejších dungeonů úrovně 14 v našem datasetu. Jinými slovy se jedná o velice dobrý čas.

Z tohoto záznamu můžeme vyvodit, že ačkoliv sem tam došlo k nějaké chybě, nebylo jich moc. Sem tam hráč mohl udělat velikou chybu, která ho stála život, ale to může být způsobeno agresivním stylem hráčů této skupiny, kteří se snaží projít dungeon co nejrychleji a snaží se bojovat proti více skupinám NPC najednou, což dostává skupinu pod větší tlak - musí se vyhýbat více mechanikám naráz, tank dostává větší damage, ale jsou poté schopni působit poškození všem NPC najednou a postupovat tak rychleji. Z času a z damageDoneScore však vyplývá, že žádná smrt skupinu nějak výrazně nezpomalila.

Na obrázku 5.7 níže můžeme vidět, kde se tento vzorek nachází mezi ostatními daty. Probíraný vzorek je znázorněn zeleně.



Obrázek 5.7. Vizualizace probíraného záznamu SD 14 vůči ostatním záznamům

5.12.2 Spires of Ascension 12 - deplete

- interruptScore : 49
- damageDoneScore : 19677334
- deathsScore : 21
- damageReceivedScore : 9972337
- failureDamageScore : 1241243
- failureDamageRatio : 0.12446861753669175
- healingScore : 9187409

- `dungeonLevel` : 12
- `dungeonName` : Spires of Ascension
- `completionTimeRatio` : 1.1896286324786325

`InterruptScore` v tomto případě sedí přesně na 75 percentil. Mezi jinými záznamy *Spires of Ascension* si vede výrazně hůř. Průměr záznamů *Spires of Ascension* je 33 a tento záznam s hodnotou 49 sedí vysoko nad 75 percentilem, který je roven hodnotě 39.5. Tohle svědčí o tom, že skupina neplnila dobře interruptování kouzel v tomto dungeonu a dostala tak zbytečně mnoho poškození. Toto je potenciálně důvod, proč byl tento *dungeon run* neúspěšný.

`DamageDoneScore` této skupiny je opravdu nízký, sedí pod 25 percentilem, který má u tohoto atributu hodnotu 21850330. Toto svědčí o pomalém postupu skupiny.

`DeathsScore` 21 je veliké číslo, 75 percentil je 18.25, v tomto dungeonu se hodně umíralo a skupinu to výrazně zpomalovalo, o čemž svědčí také nízké `DamageDoneScore`.

`DamageReceivedScore` je v tomto případě překvapivě malé. Tým neinkasoval v průběhu dungeonu mnoho poškození. Hodnota 9972337 se nachází lehce na 25 percentilem 9858500. Toto může být však dáno stylem hraní tanka, který se snaží poškození zamezit, než se s ním vyrovnat.

`FailureDamageScore` je naopak poměrně vysoké, nachází se nad mediánem, který je 1084662 a blíží se spíš hranici 75 percentil s hodnotou 1389479. Tohle nám říká, že hráči dostali poměrně vysoký damage, kterému se mohli vyhnout.

`FailureDamageRatio` je v tomto případě opravdu špatné, nachází se těsně pod hranicí 75 percentilu o hodnotě 0.124634. Tohle svědčí o tom, že se hráči nevyhýbali mechanikám moc dobře, což mohlo vést až k jejich smrti.

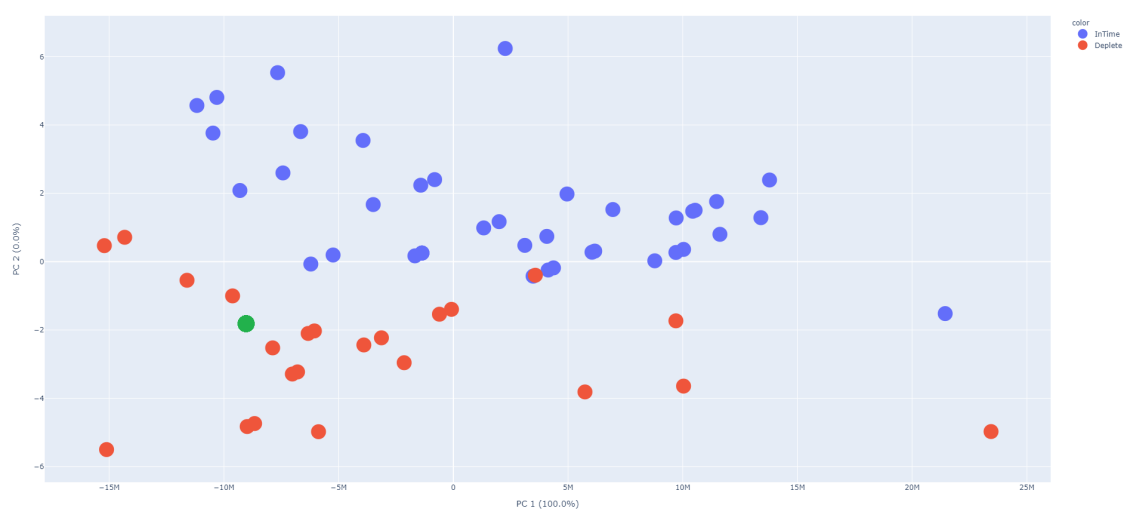
`HealingScore` se v tomto případě nachází mezi 25 percentilem a mediánem. Tohle není nijak zářivý výkon, healer této skupiny nestíhal vyléčit hráče po jejich chybách, kterých oni udělali opravdu hodně.

`DungeonLevel` tohoto záznamu je 12, což není nijak závratně vysoký klíč, ale pokud si tým nedává pozor, ignoruje mechaniky, nebo nemá dostatečné vybavení, snadno se může stát, že jej nedokončí včas. V našem datasetu se nachází 8 záznamů o dungeonech úrovně 12, z toho 5 úspěchů a 3 neúspěchy, což nám dává 62.5% úspěšnost na úrovni 12 v našem datasetu.

`CompletionTimeRatio` toho dungeon je 1.1896286324786325, což je nejhorší čas v ze všech dungeonů typu *Spires of Ascension*.

Tato skupina byla špatná ve více ohledech, hodně umírala, protože špatně nedodržovala mechaniky dungeonu a healer v této skupině nepodal dost dobrý výkon na to, aby všechny tyto chyby vybalancoval. Tohle se promítlo i na rychlosti průchodu tímto dungeonem, kdy `DamageDoneScore` této skupiny je velice nízký.

Na obrázku 5.8 níže můžeme vidět, kde se tento vzorek nachází mezi ostatními daty. Probíraný vzorek je znázorněn zeleně.

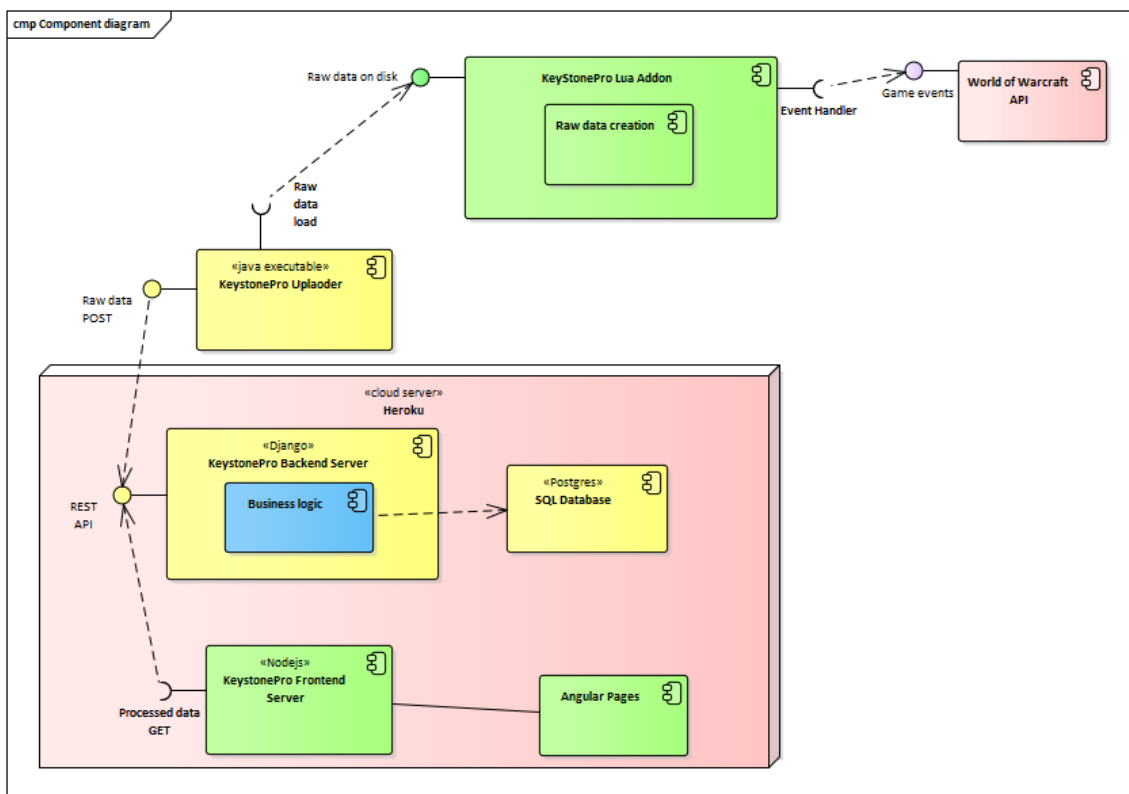


Obrázek 5.8. Vizualizace probíraného záznamu SOA 12 vůči ostatním záznamům

Kapitola 6

Návrh a implementace

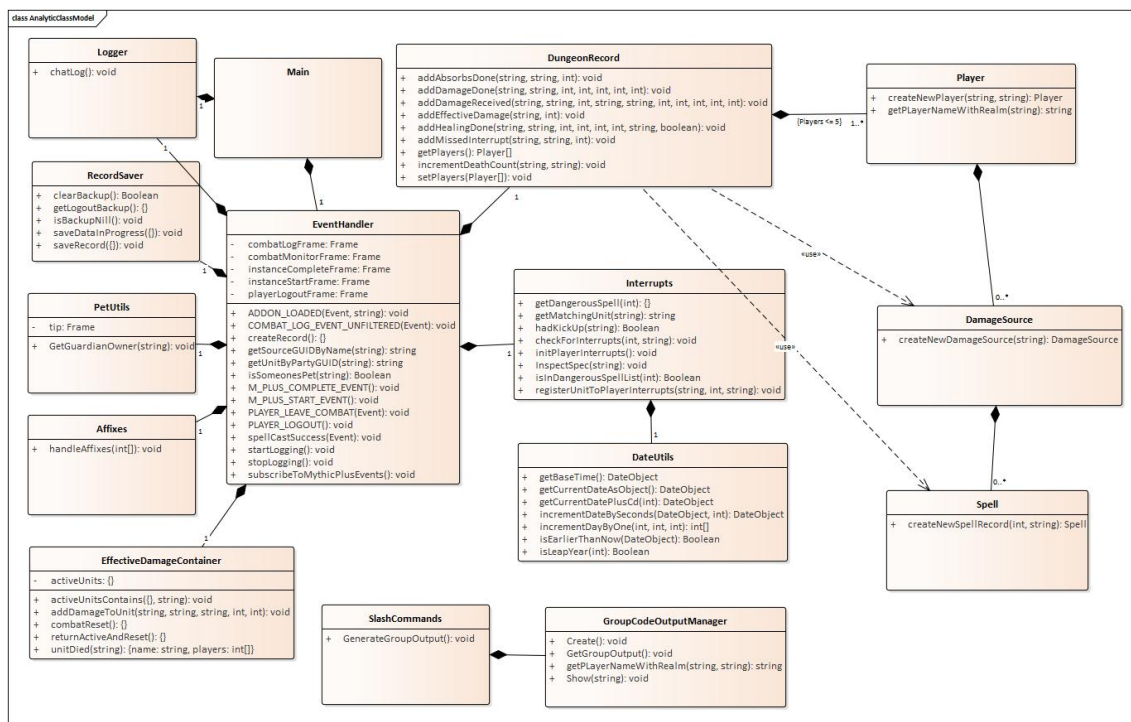
Náš systém se skládá z addonu naprogramovaného v jazyce Lua, nacházejícího se na hráčově počítači, KeystonePro uploader aplikace naprogramované v Javě, nacházející se na hráčově počítači a komunikující se serverem přes REST API, KeystonePro Backend serverem, naprogramovaného v Pythonu a Python frameworku Django, obsahujícím byznys logiku, deployovaném na Heroku, databáze PostgreSQL, deployované na Heroku, a webové aplikace sloužící jako frontend systému, naprogramované v Angularu a deployované také na Heroku, která komunikuje s backendovou částí pomocí REST API. Tohle vše je znázorněno na diagramu komponent 6.1 níže. Části implementované mnou jsou znázorněny zelenou barvou, části implementované Martinem jsou znázorněny žlutě a části na kterých jsme se podíleli oba jsou znázorněny modře.



Obrázek 6.1. Diagram komponent našeho systému [10]

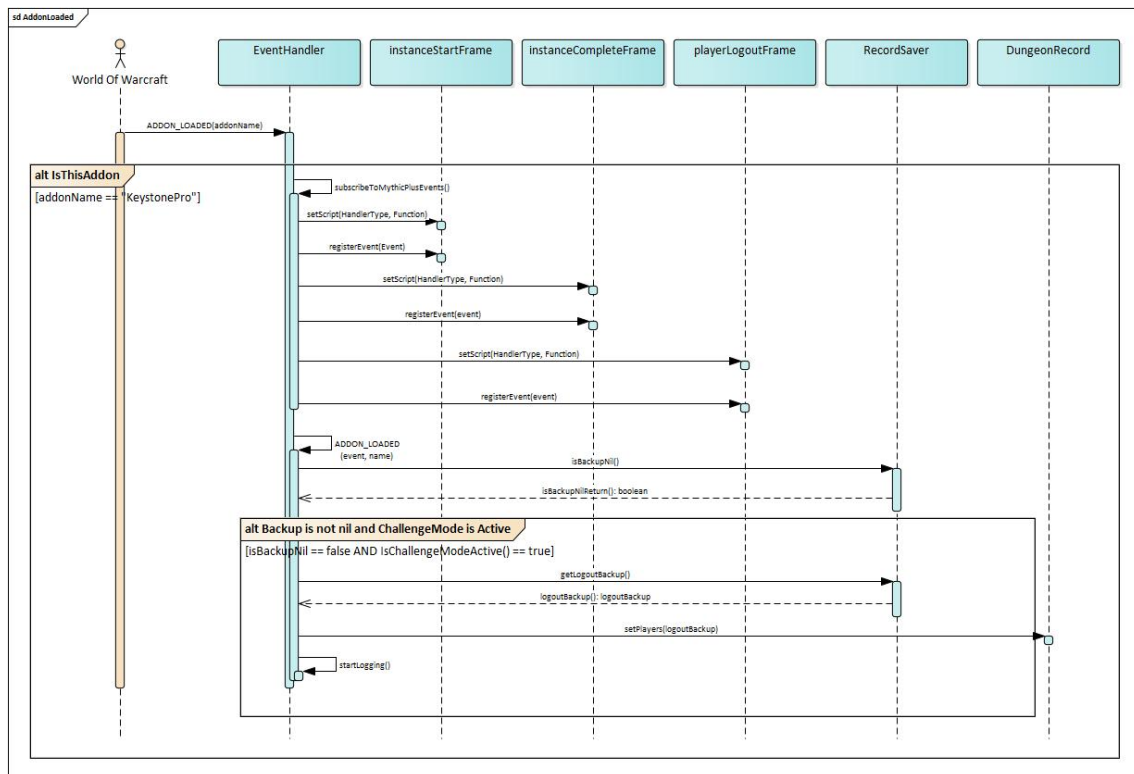
6.1 Herní addon

Herní addon se skládá z několika částí, kde každá odděluje nějakou část logiky. Jeho struktura a rozdělení logiky do funkcí jsou popsány na analytickém modelu tříd 6.2. Na něm můžeme vidět, že hlavní část dění se odehrává v souboru EventHandler.



Obrázek 6.2. Analytický model tříd addonu

Soubor EventHandler v sobě obsahuje logiku, která reaguje na stěžejní eventy hry. Hned co se hráč naloguje do hry, tak se mu při načítání hry načtou i jeho addony. Za každý addon potom hra emituje event ADDON_LOADED. Addon KeystonePro na tenhle event také reaguje. Logika, která se odehraje při emitování tohoto eventu, je zachycena v následujícím diagramu 6.3. Nejdříve proběhne kontrola, event ADDON_LOADED byl vyvolán na addon KeystonePro. Pokud ano, pak se v souboru EventHandler zavolá funkce subscribeToMythicPlusEvents(), ve které se addon přihlásí k odběru eventů CHALLENGE_MODE_START, CHALLENGE_MODE_COMPLETED, ADDON_LOADED a PLAYER_ENTERING_WORLD. Poté se spustí funkce ADDON_LOADED a proběhne kontrola, zda-li je backup dat nil. Pokud backup dat není nil a zároveň se hráč nachází v aktivním dungeonu módu Mythic+, obnoví se záznam dungeonu a nové záznamy se připojí do backupu dat.



Obrázek 6.3. Sekvenční diagram při načtení addonu

6.1.1 Důležité herní eventy

Event `CHALLENGE_MODE_START` signalizuje start Mythic+ dungeonu a spustí sbírání dat. Event `CHALLENGE_MODE_COMPLETED` signalizuje dokončení Mythic+ dungeonu a připraví datovou strukturu s datem dokončení, obtížností, celkovým časem, typem dungeonu, záznamy hráčů a dalšími atributy. Dále probíhá subscribe k eventu `ADDON_LOADED` kvůli zálohování dat. Sbíraná data jsou během dungeonu uchovávána v operační paměti addonu. Když se hráč v průběhu Mythic+ runu odloguje z postavy, nebo z dungeonu vyjde ven, dojde při načítacích obrazovkách k znovunačtení addonu a tím pádem se resetuje jeho paměť. Aby nedošlo ke ztrátě dat, ukládá addon data o průběhu dungeonu do saved variables (viz. dále). Event `ADDON_LOADED` je emitovaný právě tehdy, když jsou tyto saved variables připravené k načtení. Dále probíhá subscribe k eventu `PLAYER_ENTERING_WORLD`. Tento event je emitován právě tehdy, když hráč vyjde ven z dungeonu. Mimo dungeon se mohou nacházet jiní hráči, kteří mohou svými akcemi poškodit a znehodnotit sbíraná data addonem. Například 2 hráči přímo před vchodem do dungeonu bojují proti sobě. Addon by normálně jejich akce zaznamenal a zapsal do právě spuštěného dungeon záznamu. Právě proto se v eventu `PLAYER_ENTERING_WORLD` kontroluje, zda-li je v daném světě aktivní Mythic+ run, a v případě že není, vypíná se záznam. Na obrázku můžeme v 6.4 vidět funkci, která se spustí při eventech `PLAYER_ENTERING_WORLD` a `ADDON_LOADED`.

```

function playerLogoutFrame:ADDON_LOADED[event, name]
    if event == "ADDON_LOADED" then
        if name ~= ADDON_NAME then return end

        playerLogoutFrame:UnregisterEvent("ADDON_LOADED")
        if C_ChallengeMode:IsChallengeModeActive() and not RecordSaver:isBackupNil() then
            DungeonRecord:setPlayers(RecordSaver:getLogoutBackup())
            EventHandler:startLogging()
        end
    elseif event == "PLAYER_ENTERING_WORLD" then
        if not C_ChallengeMode:IsChallengeModeActive() then
            EventHandler:stopLogging()
        end
    end
end
end

```

Obrázek 6.4. Kód obsluhující event ADDON_LOADED a PLAYER_ENTERING_WORLD

Nejdůležitější event, který nám poskytuje veškerá data, která sbíráme, je COMBAT_LOG_EVENT_UNFILTERED. K tomuto eventu se subscribuje při spouštění logování, tudíž při emitování eventu CHALLENGE_MODE_START. Tento event je opravdu robustní a nese v sobě spoustu sub eventů a různých informací - v některých případech může mít až 24 argumentů. Například hráč použije abilitu “Backstab”, která dá hráčově cíli poškození. V argumentech eventu se potom nachází kdo dal komu kolik damage a čím, kolik damage bylo absorbováno, kolik damage byl takzvaný overkill (např. hráč dal více damage než měl jeho cíl hit pointů) a jiné informace. Tyto informace jsou poté zpracovány ve funkci combatLogFrame:COMBAT_LOG_EVENT_UNFILTERED() v souboru EventHandler.lua, která poté rozdělí tok těchto dat dále do souboru DungeonRecord.lua, kde se nachází logika k zaznamenání dat do datové struktury odpovídající dungeon runu.

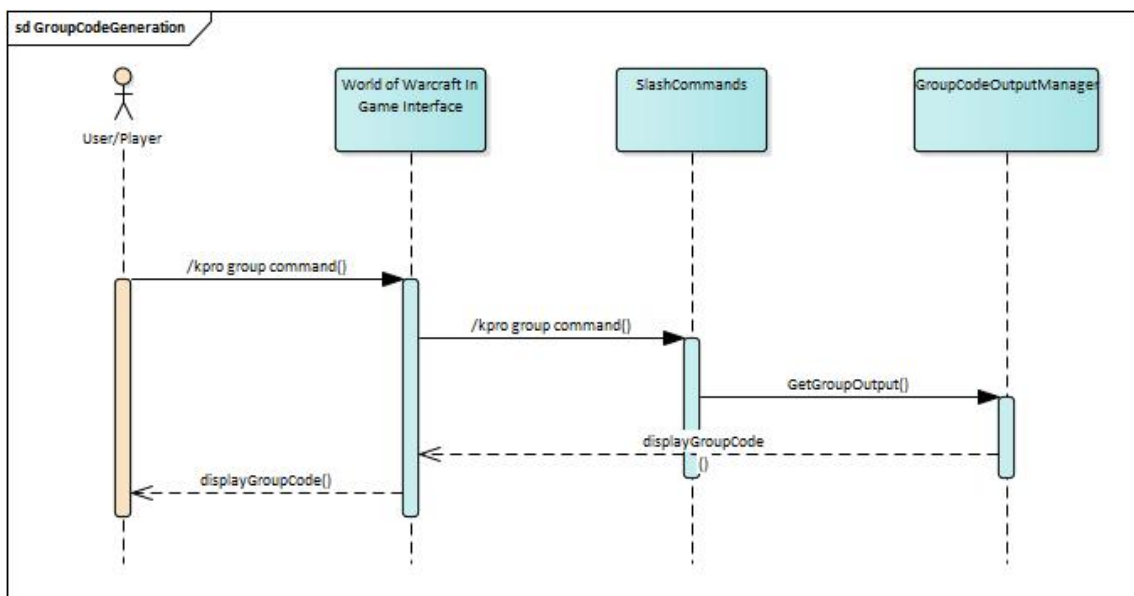
Na odkazu v sekci literatura se nachází komunitní dokumentace k tomuto eventu. [38] Na obrázku A.1 příloze můžeme vidět sekvenční diagram, ve kterém je znázorněn průběh zpracování eventu COMBAT_LOG_EVENT_UNFILTERED v addonu.

Hlavní interakce uživatele s addonem spočívá ve vygenerování kódu skupiny pro webovou aplikaci systému. Uživatel zadá do herního chatu, který slouží zároveň jako konzole, příkaz, který poté zobrazí kód v okně na obrazovce pro zkopírování.



Obrázek 6.5. Uživatel si příkazem nechal vygenerovat kód skupiny [4]

Na obrázku 6.5 můžeme vidět, jak se kód skupiny zobrazí bezprostředně po zadání příkazu.



Obrázek 6.6. Sekvenční diagram generace kódu skupiny

Na tomto sekvenčním diagramu 6.6 můžeme vidět, jak v kódu addonu probíhá zobrazení kódu skupiny. Uživatel zadá příkaz `/printraid` do herního chatu. World of Warcraft interface potom zavolá registrovaný příkaz naším addonem, ve kterém se jen zavolá logika souboru `GroupCodeOutputManager`, která poté vykreslí požadovaný output na obrazovku.

Hlavní konfigurace addonu se nachází v `.toc` souboru umístěném přímo ve složce s ostatními `.lua` soubory. Nastavuje se tam mimo jiné jméno addonu, jak je poté zobrazováno ve hře, verze addonu, patch hry pro který je verze určená, závislé moduly

a addony (dependencies), saved variables (viz. dále), které skripty .lua skripty se při načtení addonu mají spustit a v jakém pořadí a další nepovinné konfigurační věci.

```

KeystonePro.toc
1  ## Interface: 90002
2  ## Title: Keystone Pro
3  ## Version: 1.0.0
4  ## Notes: Testovací addon
5  ## Author: Lister
6  ## DefaultState: disabled
7  ## SavedVariables: RecordedDungeon, LogoutBackup
8
9  Logger.lua
10 EffectiveDamageUnitContainer.lua
11 Spell.lua
12 DamageSource.lua
13 Player.lua
14 DungeonRecord.lua
15 RecordSaver.lua
16 PetUtils.lua
17 DateUtils.lua
18 Interrupts.lua
19 Affixes.lua
20 EventHandler.lua
21 GroupCodeOutputManager.lua
22 SlashCommands.lua
23 Main.lua
24
25 libs\WoWUnit-master\Interface\Frames.xml
26 libs\WoWUnit-master\Interface\Panel.lua
27
28 libs\WoWUnit-master\Classes\Group.lua
29 libs\WoWUnit-master\Classes\Test.lua
30
31 libs\WoWUnit-master\WoWUnit.lua
32
33 UnitTests.lua

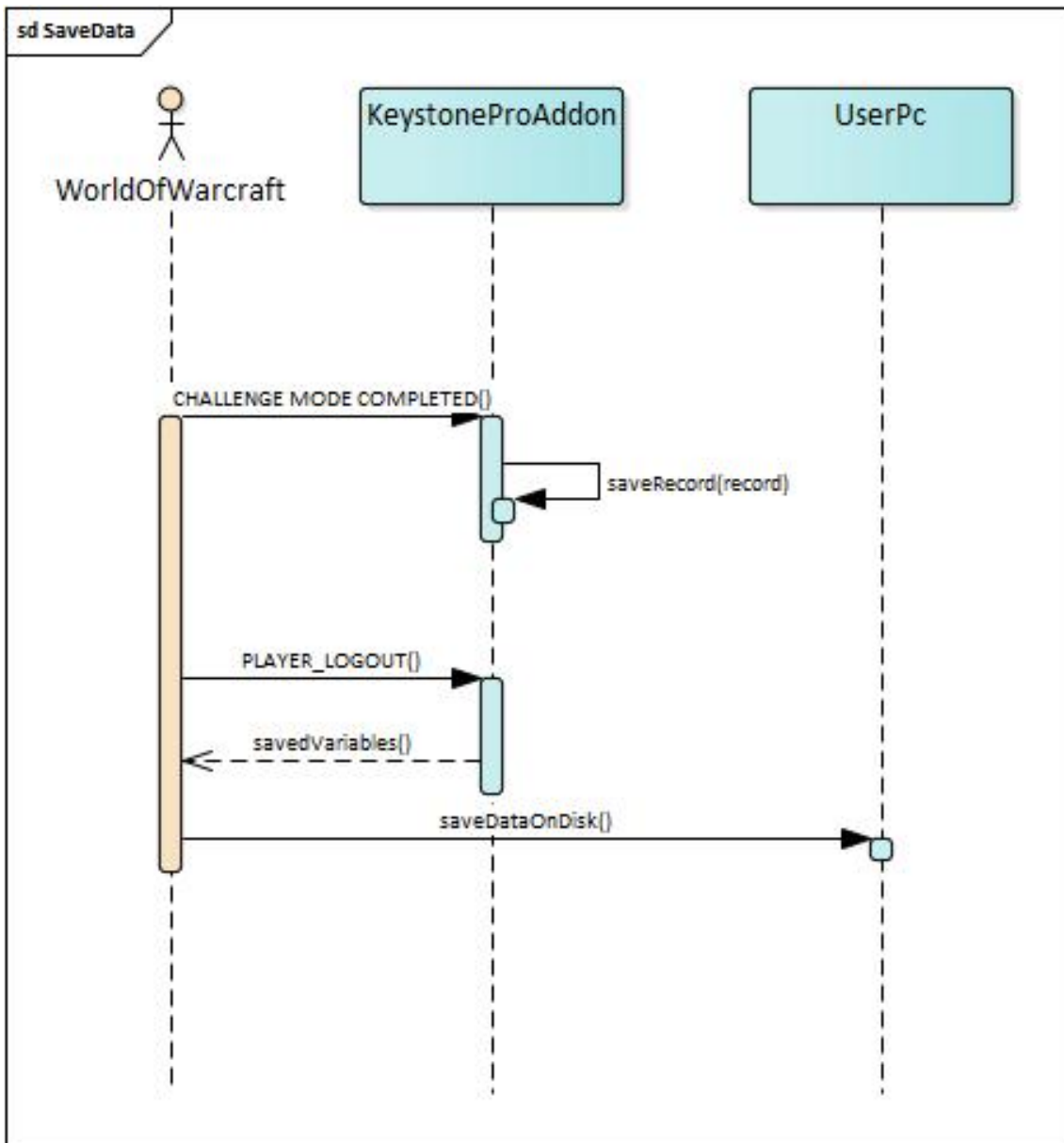
```

Obrázek 6.7. .toc soubor addonu

Na tomto obrázku 6.7 můžeme vidět KeystonePro.toc soubor našeho addonu.

6.1.2 Ukládání dungeonu na disk

Původní idea byla taková, že addon bude přímo ukládat záznamy na disk. Blizzard však zápis na disk blokuje a třídy/části jazyka lua obsahující logiku se zápisem na disk jsou blokovány a neinterpretují se herním interpreterem. Na druhou stranu blizzard pro ukládání nastavení addonů poskytuje systém saved variables. To jsou proměnné které se specifikují v .toc souboru addonu. Saved variables se mohou vázat na konkrétní postavu hráče, nebo na celý hráčův účet. Tyto proměnné jsou potom globálně dostupné v celém programu a při odlogování hráče ze hry se jejich obsah uloží do složky /World of Warcraft/_retail_/WTF/Account/"kód účtu"/ a dále do /SavedVariables/"jméno addonu kterému patří" pokud jsou společné pro všechny postavy a vážou se na celý účet nebo /"server na kterém je postava uložena"/"jméno postavy"/SavedVariables/"jméno addonu" pokud se vážou na jednotlivé postavy na účtě. Tento ukládací proces můžeme vidět na diagramu 6.8 níže. Do saved variables jsou ukládat jen lua jazykem poskytnuté datové typy (string, number, table aj.). Při naložování addonu se jejich obsah načte z disku a nahraje do paměti addonu.



Obrázek 6.8. Sekvenční diagram ukládání záznamů na disk

6.2 Frontend

Frontend je naprogramovaný ve frameworku Angular verze 11.2.6 za pomoci Angular CLI verze 11.2.6.. Pro moderní vzhled aplikace byla použita knihovna Angular-Material. [39]

6.2.1 Blizzard API a Oauth

Blizzard API poskytuje nejrůznější informace o stavu hry nebo o profilech a postavách jednotlivých hráčů. Skrze něj může programátor například zjistit, jaké profese ovládají různé postavy hráče, jaké mají vybavení, čeho ve hře dosáhli zobrazením jejich achievementů a různých statistik. Kromě údajů o postavách a hráčích, které se vážou na konkrétní profily, lze v blizzard API zjistit jaké vybavení se právě nachází v aukci, informace a ikonky k různým předmětům a vybavení ve hře a v neposlední řadě i k sezónním

herním systémům nebo mechanismům jako jsou například Covenanty v současné době. K přístupu k blizzard API je potřeba zaregistrovat se na stránce [40]. Programátor poté získá přístup ke client ID a client secret. Aby aplikace mohla provádět konkrétní requesty na blizzard API je třeba vygenerovat Oauth token. Ten se generuje POST requestem na blizzard API, ve kterém je mimo jiné třeba do parametrů uvést client ID a client secret. Poté může programátor vygenerovaný Oauth token přidat do parametrů různých get requestů a využívat tak blizzard API, v našem případě k načítání avatarů hráčů a výpočtu jejich item levelu. Oauth token používá lazy initialization, to znamená, že se kontrola jeho existence a případný POST request provádí až při přímém zadání dat od uživatele a těsně před odesláním requestů na blizzard API pro data. Spolu s Oauth tokenem přichází i jeho expirační doba, po které již nebude dále platný. Tuto expirační dobu uchovávám a pomocí třídy Date a kontroluji zda-li neuplynula. Pokud ano, požádá frontend o nový token, pokud ne, vrátí starý.

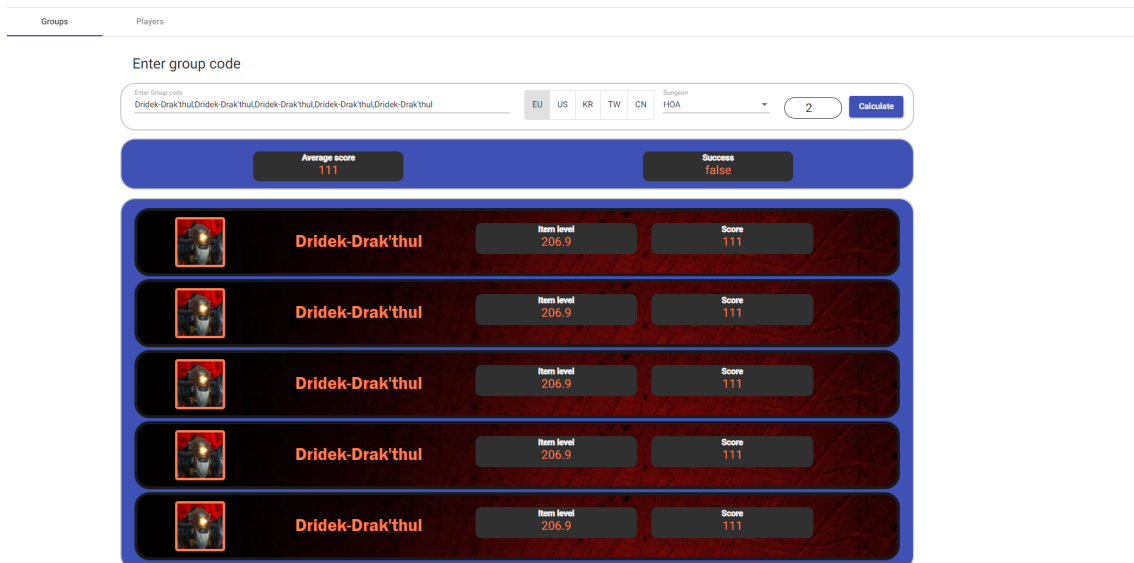


Obrázek 6.9. Achievmenty hráče zobrazené ve hře [4]

Na obrázku 6.9 vidíme interface s achievmenty postavy. Údaje v něm obsažené jsou získatelné také přes blizzard API.

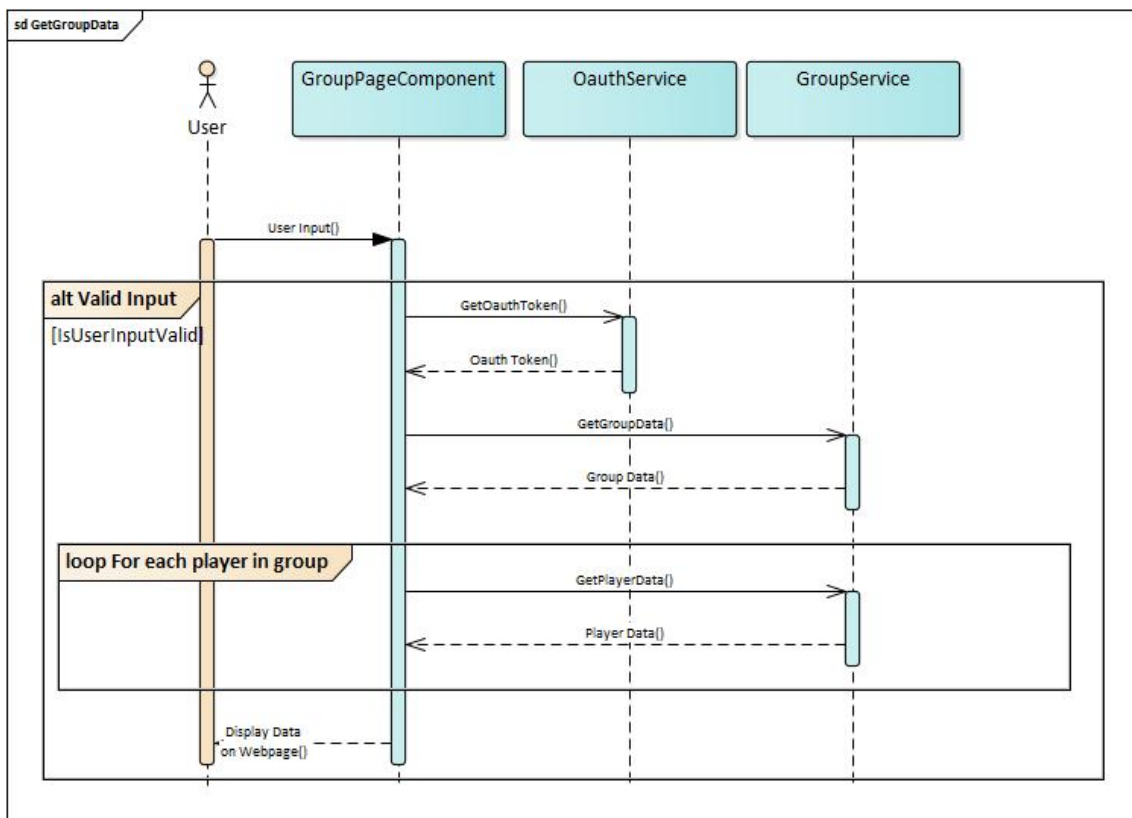
6.2.2 Interakce s uživatelem a servisní vrstva

Pro komunikaci s uživatelem slouží dvě angular komponenty GroupPage a PlayerPage. GroupPage poskytuje rozhraní pro získání predikce pro pětici postav hráčů. PlayerPage poskytuje rozhraní k zobrazení jednotlivých hráčů a jejich skóre. Na obrázku 6.10 můžeme vidět GroupPage po zadání skupiny obsahující 5x postavu Dridek z realmu (herního serveru) Drak'thul.

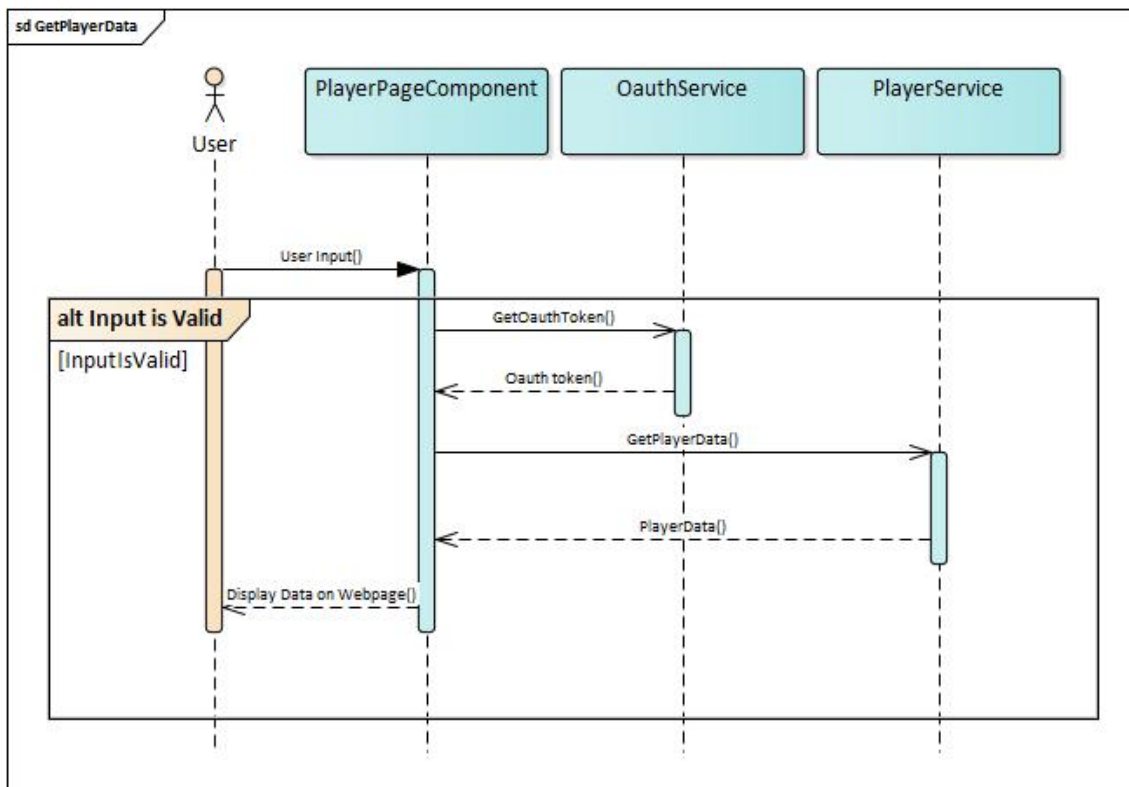


Obrázek 6.10. Vyhledání skupiny o složení 5x Dridek na frontendu systému

Každá z těchto komponent je obsluhována jednou servisou, která je plní komponenty daty. Každá z těchto dvou service plní jeden usecase. Každý z obou usecasů je popsán následujícími sekvenčními diagramy. Na obrázku 6.11 vidíme zpracování uživatelského inputu pro GroupPage. Na obrázku 6.12 vidíme zpracování uživatelského inputu do PlayerPage.



Obrázek 6.11. Sekvenční diagram user requestu na data o skupině



Obrázek 6.12. Sekvenční diagram user requestu na data o postavě

V metodách `getGroupData` a `getPlayerData` probíhají dva asynchronní requesty v každé. Jeden request má za úkol donést data z našeho systému - skóre hráče/hráčů a v případě skupiny i predikci. Druhý request směřuje na blizzard API, odkud donáší avatary hráčů a seznam jejich vybavení, ze kterého se poté počítá item level. Všechny tyto atributy jsou součástí jednoho angular interface a jsou potom součástí jednoho objektu. Proto je důležité, aby asynchronně pracující service k těmto dvěma requestům přistupovala jako k jednomu. Toho je docíleno použitím modulu `rxjs` a jeho operátorů `MergeMap` a `Map`, které sjednotí dvě `Observable` dvou různých requestů do jediného a data ze subscriptionu této sjednocené `Observable` přijdou až poté, co frontend obdrží obě response na oba requesty.

Na obrázku 6.13 můžeme vidět použití `rxjs` operátorů v metodě `getPlayerData` v servise `player.service.ts`.


```

27 const itemsAddress = "https://" + region + ".api.blizzard.com/profile/wow/character/" + realm + "/" + charName +
28 "/equipment?namespace=profile-" + region +
29 "&locale=en_GB&access_token=" + apiKey
30
31 const $itemsObservable = this.httpClient.get(itemsAddress)
32 const $mediaObservable = this.httpClient.get(mediaAddress) as Observable<BlizzMedia>
33
34 const $blizzardDataObs = $mediaObservable.pipe(
35   mergeMap( blizzMedia => {
36     console.log(blizzMedia)
37     return $itemsObservable.pipe(
38       map( items => {
39         return {items: (items as {equipped_items: any}).equipped_items, media: blizzMedia}
40       })
41     )
42   })
43 )
44 const $playerScore = this.httpClient.get(scoreAddress)
45
46 return $blizzardDataObs.pipe(
47   mergeMap( blizzData => {
48     return $playerScore.pipe(map(
49       scoreObj => {
50         return {
51           playerName: blizzData.media.character.name + "-" + blizzData.media.character.realm.name,
52           score: (scoreObj as {score: number}).score,
53           image: (blizzData.media as BlizzMedia).assets[0].value,
54           itemLevel: Utils.getItemLevelFromBlizzardItems(blizzData.items)
55         }
56       })
57     ))
58   })),
59 )
60

```

Obrázek 6.13. Kód metody get Player, použití rxjs operátorů

6.3 Implementace Algoritmů Predikce

Jako implementaci algoritmů jsem se rozhodl použít kódy ze svých domácích úkolů do předmětu BE5B33RPZ Pattern Recognition and Machine Learning, který jsem absolvoval v zimním semestru roku 2020/2021. Všechny kódy jsou v jazyce Python a používají knihovnu Numpy pro manipulaci s daty a matematické operace. Kódy vycházejí z šablony poskytnutého předmětem a kódy pro SVM a Adaboost v sobě obsahují předprogramované pomocné metody. Na obrázku 6.14 vidíme kód použitý v systému.

```

n = len(y)
sample_weights = np.zeros(n)

sample_weights[y == 1] = 0.5 / len(y[y == 1])
sample_weights[y == -1] = 0.5 / len(y[y == -1])

error_hist = np.zeros(num_steps)
upper_bound = np.zeros(num_steps)
strong_classifier = {'wc': [], 'alpha': np.zeros(num_steps)}

for i in range(num_steps):
    best_weak, error = find_best_weak(X, y, sample_weights)

    if error >= 0.5:
        return None, None, None

    if error == 0:
        strong_classifier['wc'].append(best_weak)
        strong_classifier['alpha'][i] = 1
        return strong_classifier, None, None

    error_hist[i] = error
    strong_classifier['wc'].append(best_weak)
    strong_classifier['alpha'][i] = 0.5 * np.log((1-error)/error)

    coef = np.sign(best_weak['parity'] * (X[best_weak['idx']] - best_weak['theta']))

    sample_weights[y == coef] *= math.sqrt(error/(1-error))
    sample_weights[y != coef] *= math.sqrt((1-error)/error)

    Z = np.sum(sample_weights)
    upper_bound[i] = Z * np.prod(upper_bound[i-1:i])
    sample_weights /= Z

strong_classifier['wc'] = np.array(strong_classifier['wc'])
return strong_classifier, error_hist, upper_bound

```

Obrázek 6.14. Kód adaboostu použitý v systému

Nejdříve jsem použil perceptron. Data v datasetu se však ukázala být lineárně neseparabilní, a proto algoritmus ani nedokonalgoval. Jako další jsem vyzkoušel SVM s RBF kernelem. SVM však nedokázalo dobře odhadnout podobu dat a i po iteraci přes parametry SVM byla všechna data v testovací sadě klasifikována jednotně, buď jako neúspěchy nebo úspěchy.

S použitím algoritmu Adaboostu jsme začali dostávat smysluplné výsledky. Jeden prvek datasetu představuje jeden běh dungeonem v módu Mythic+. Cílem v této části bylo natrénovat klasifikátor tak, aby rozpoznal úspěšně dokončený dungeon včas od neúspěšného dokončení po vypršení času - depletu. Pro trénování jsem rozdělil dataset v poměru 48:16 na trénovací sadu a testovací sadu. Obměňováním složení trénovací a testovací sady se error pohyboval od 14% po 25%. Tyto velké změny v erroru má na svědomí nedostatek dat, a proto natrénování klasifikátoru záleží na jednotlivém vzorku dat více, než by mělo. Na finální testovací sadě jsem dostal celkový error 18,75%, z toho 0% na třídě úspěch a 37,5% na třídě neúspěch/deplete. Toto rozdělení sady jsem zvolil proto, protože poměr úspěchů a neúspěchů v reálném prostředí zhruba odpovídal poměru v testovací sadě.

Takto natrénovaný klasifikátor jsme poté umístili na server KeystonePro, pro predikci úspěchu týmů. Klasifikátor poté bude predikovat, zda-li průměrný výkon hráčů v zadaném týmu bude stačit na dungeon zvolené úrovně.

6.4 Implementace výpočtu skóre

Návrhu výpočtu a poté i samotné implementaci skóre se věnoval a podrobněji věnuje ve své práci Martin Lukeš. Martin se rozhodl porovnávat zvláště úspěšnost v určitých oblastech hráčů mezi sebou a vytvořit tím percentily. To znamená, že pokud má hráč percentil 93.0, tak 7 procent hráčů je lepších než on a 93 procent hráčů je horších v dané oblasti. V některých oblastech je nesmyslné porovnávat skóre vůči jiným rolím. Například HEALER nedá nikdy takové poškození jako DAMAGE a naopak DAMAGE nikdy nevyлéčí tolik jako HEALER. Proto by se výpočet skóre dal shrnout jako percentilní počet v různých oblastech s ohledem na herní role. Celkové skóre je průměr těchto percentilů. [10]

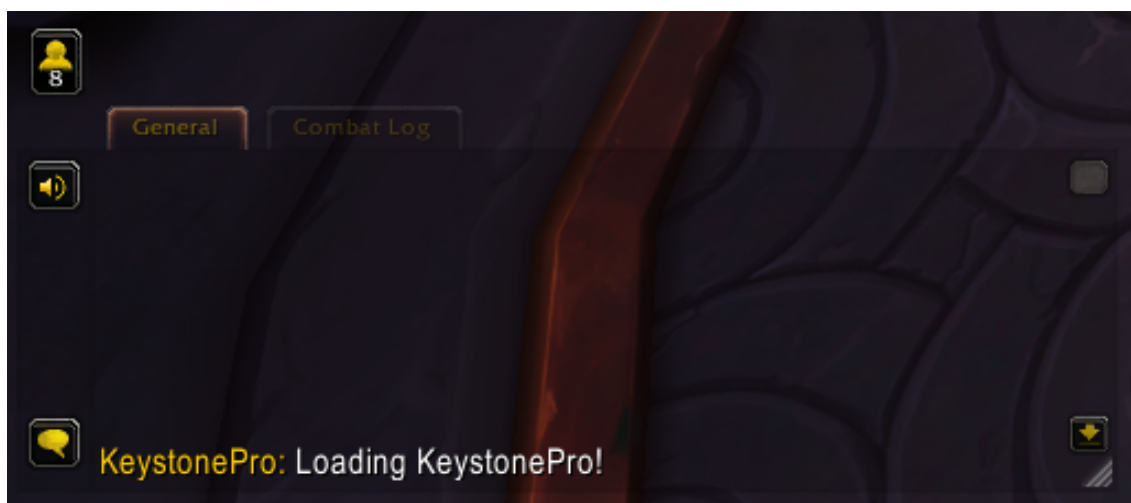
7.1.2 Buggy v průběhu vývoje addonu

- Občas se v záznamech objevovali hráči navíc, což byl důsledek špatného vypínání /nevypínání záznamu, při opuštění dungeonu. Když cizí hráči prováděli nějaké akce přímo před dungeonem, addon tohle chování zaznamenal. Bylo přidáno vypínání logování když hráč opustil dungeon a zapínání logování, když se do něj vrátil v průběhu Mythic+ módu.
- Docházelo ke špatnému formátování při ukládání více dungů po sobě, vkládání jednoho záznamu do dalšího. Tohle byl důsledek staré verze, která podporovala uložení pouze jednoho dungeonu. S novou verzí se místo jednoho dungeonu začal ukládat table (datová struktura jazyku Lua) dungeonů. Jelikož záznam dungeonu je také table, tak volání metody append na záznamy dungeonů neházel žádný error. Muselo se vyřešit manuální úpravou struktury souboru ve složce SavedVariables, kterou používáme pro ukládání dungeonu na disk.
- V dungeonu Mists of Tirne Scithe měli všichni hráči příliš špatné interrupt skóre. Tohle bylo způsobeno tím, že některá NPC bojují v dungeonu proti sobě a u toho čarují kouzla, která jsou označená jako nebezpečná - tudíž jejich čarování musí být přerušeno. Takže stačilo aby tým se nacházel poblíž těchto NPC, a i přesto když s nimi nebojoval, addon zaznamenal nepřerušená čarování těchto NPC a připsal chybu všem hráčům, kteří měli připravený interrupt spell. Bug byl vyřešen, zjistilo se, že onen nebezpečný spell má v blizzard databázi dvě IDčka, jedno pro čarování spellu na hráče, druhé pro NPC. ID pro čarování na NPC bylo odstraněno.

7.1.3 Uživatelské testování addonu

Addon samotný nemá s uživatelem nijak interagovat při sběru dat. Tudíž jeho hlavní úloha pro interakci s uživatelem je nezpomalovat hru a vůbec se během ní neprojevat. Podle uživatelů bylo těžké ověřit, zda-li addon vůbec běží a jestli jej mají vůbec správně zapnutý. Proto jsem se rozhodl přidat do addonu výpisy do herního chatu, který slouží také jako konzole, viditelné pouze pro vlastníka addonu. První výpis se objeví při nalogování hráče do hry na postavu a naložování addonu. Druhý výpis se objeví při spuštění dungeon runu a poslední se zobrazí při jeho dokončení.

Na obrázku 7.2 můžeme vidět výpis addonu do chatu při jeho naložování.



Obrázek 7.2. Výpis addonu do chatu při jeho naložování [4]

7.2 Testování frontendu

Angular má na unit testy a integrační testy vestavěný framework Jasmine. Jasmine je Behavior Driven vývojově testovací framework pro Javascript. Nezávisí na prohlížečích, DOM, nebo jakémkoli jiném Javascriptovém frameworku. Proto je vhodná pro webové stránky, Node.js projekty, nebo prostě kekoliv, kde může běžet Javascript. [42] Na spouštění testů v prohlížeči má potom Angular vestavěný další framework jménem Karma. Karma vytvoří umělý server a na něm potom rozběhne dané testy v uživateli zvoleném prohlížeči.

Karma je v zásadě nástroj, který vytvoří webový server, který spustí zdrojový kód oproti testovacímu kódu za každý připojený prohlížeč. Výsledky každého testu pro každý prohlížeč jsou zkoumány a zobrazeny skrze příkazový řádek programátorovi a to tak, že mohou vidět které testy pro jaké prohlížeče prošly nebo nepošly.

Prohlížeč může být spuštěn:

- manuálně, navštívením URL adresy, kde poslouchá Karma server (typicky `http://localhost:9876`)
- nebo automaticky, nastavením Karmy tak, aby věděla které prohlížeče spustit při běhu testů

Karma také hlídá všechny soubory, specifikované v konfiguračním souboru a kdykoliv se některý z nich změní, všechny testy se rozběhnou znovu posláním signálu na testovací server, aby informoval všechny zaznamenané prohlížeče, aby znovu rozběhly testovací kód. Každý z těchto prohlížečů poté načte zdrojové soubory do IFramu, spustí testy a reportuje výsledky zpátky serveru.

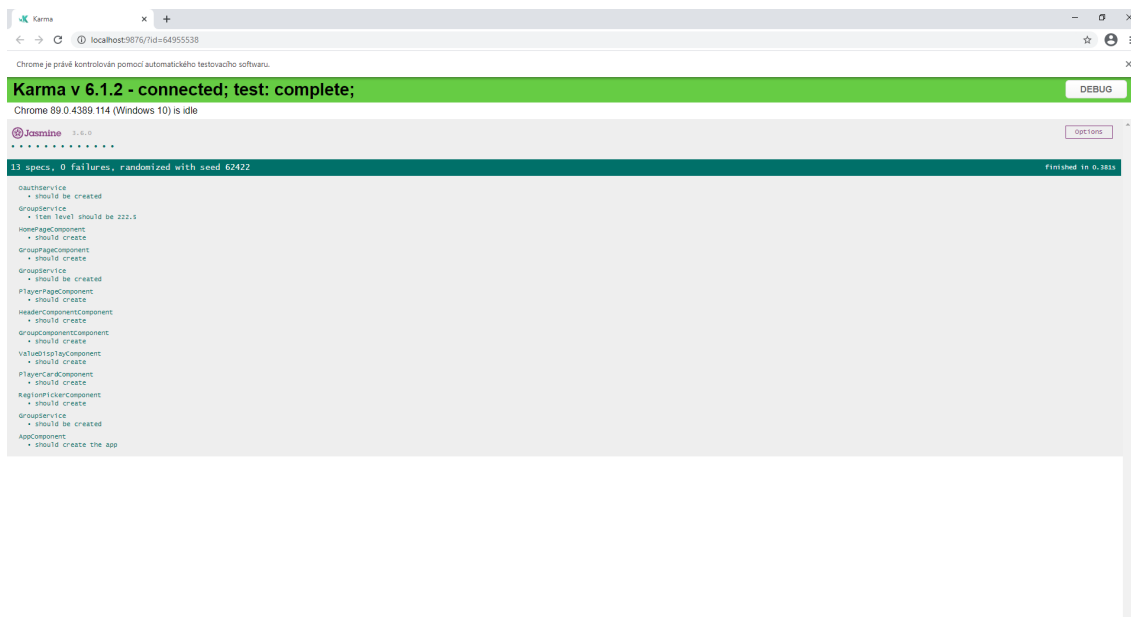
Server posílá výsledky ze všech zaznamenaných prohlížečů a prezentuje je zpátky programátorovi.

Tohle je jen stručný přehled, protože při používání Karmy není nutné znát, jak interně funguje. [43] Karma i Jasmine jsou také stručně popsány v oficiální dokumentaci Angularu. [44]

Jako best practice je v Angularu považované používat při tvorbě projektu `angular cli` (command line interface). Při vytváření komponent, service, direktiv a jiných stavebních prvků projektu přes `angular cli` se automaticky ve složce s daným prvkem vytvoří i testovací soubor s koncovkou `spec.ts` už s předpřipraveným pro základní unit testy. Vše je tím pádem připravené a programátorovi stačí testy jen napsat. V jednom testovacím souboru s koncovkou `spec.ts` se nacházejí testovací sady (test suite) definované funkcí `describe`. Tyto sady obsahují typické metody pro testovací frameworky jako třeba `BeforeEach`, `AfterEach`, `BeforeAll`, `AfterAll`, které přispívají k dodržení principu DRY (don't repeat yourself), a také samotné testy, které jsou potom definované funkcí `it`. Každý test by měl obsahovat alespoň jedno volání funkce `expect` (funguje jako `assert` v jiných jazycích), která potom přispívá do vyhodnocení testu. Jestli jsou všechny `expecty` splněny, je test vyhodnocen jako úspěšný, je-li tomu jinak, je test prohlášen za neúspěšný. Programátor může také spouštět jen vybrané testovací sady nebo testy. Pokud test definovaný funkcí `it` přejmenuje na `xit`, tak daný test vyřadí dočasně ze sady. Pokud test definovaný funkcí `it` přejmenuje na `fit`, tak se spustí pouze tento test.

Spuštění testů potom probíhá příkazem `ng test` v kořenovém adresáři projektu.

Na obrázku 7.3 můžeme vidět uživatelské rozhraní frameworku Karma pro zobrazení úspěšných a neúspěšných unit testů.



Obrázek 7.3. Uživatelské rozhraní frameworku Karma pro zobrazení úspěšných a neúspěšných unit testů [43]

7.2.1 Uživatelské testování frontendu

Během uživatelského testování frontendu jsme narazili na bug při výpočtu item levelu. Ten původně probíhal tak, že se provedl call na blizzard API pro seznam itemů, které má zrovna daný hráč na sobě. Z těchto itemů se potom spočítal průměr. Pokud byl hráč vybavený obouruční zbraní, započítala se do průměru 2x, protože zabírá 2 sloty místo klasického jednoho. Hráč si může celkově oblétnout vybavení až do 18 slotů, z toho 16 slotů se počítá do item levelu, a zbylé 2 jsou pouze kosmetické. K chybě docházelo pouze pokud měl hráč vzácné vybavení s vysokým item levelem ve 2 zbylých kosmetických slotech. Došlo k úpravě kodu a přidání příslušného unit testu.

7.3 Testování predikce a skóre

Nasbírali jsme celkem data o 215 různých hráčích.

7.3.1 Skóre

Do top 10 našeho rankingového systému se dostalo těchto 10 hráčů.

- **Nivasboy-Zenedar** - Zřejmě se jedná o omyl, tohle není příliš zkušený hráč, nechodí vysoké klíče a jeho profil nepůsobí silným dojmem. Dle Raider.io má skóre 546, což je opravdu nízko. V našem systému má odchozený pouze 1 dungeon, takže může jít o náhodnou skvělou performanci. Odchozený dungeon je Necrotic Wake 5, což je velice nízký klíč. Navíc složení jeho týmu svoji výbavou vysoce převyšovalo tuhle úroveň, takže neměl prostor udělat moc chyb.
- **Sneakyball-Thrall** - Podle jeho profilu se jedná o velice kvalitního hráče. V podobném herním módu Raid je tento hráč v 1% nejlepších hráčů na světě, dokázal získat prestižní Cutting Edge, což znamená, že dokázal s týmem vyčistit Mythic Raid této sezóny, což se podaří tak 1% všech hráčů. Podle Raider.io má skóre 1,536, což je působivý výsledek.

- **Dudris-Thrall** - Jedná se o vedlejší postavu poměrně zkušeného hráče. Na své hlavní postavě má Raider.io skóre 1,481, což není málo. V našem systému má jeden záznam a to z klíče Halls of Atonement 8. To je poměrně nízký klíč a pro zkušeného hráče není problém tuto obtížnost zahrát téměř bez chyby.
- **Rykerros-Alonsus** - Podle Raider.io se jedná o průměrného hráče. Svoje skóre si vysloužil na De Other Side 16, což je poměrně vysoký klíč na tohoto hráče. Jeho skóre mu zvedlo HealingScore, i přesto že je to DamageDealer. Dost možná proto se nachází tento průměrný hráč takto vysoko. Na Raider.io má skóre 1,365, což je čistý průměr. Nicméně jeho jediný záznam se mu skutečně povedl, protože vyniká i v ostatních attributech a výsledné skóre si tím vůbec nekazí.
- **Sickyo-Thrall** - Podle jeho profilu se jedná o průměrného hráče. Na Raider.io má skóre 1,339, což je docela průměrné u lidí, co se módu Mythic+ pravidelně věnují. Dle hodnot v jeho záznamu se mu však zaznamenaný dungeon povedl a má v něm pěkná čísla. Jednalo se o Necrotic Wake 14, což je normální obtížnost klíče. NPC na této úrovni již něco vydrží, takže je spoustu prostoru rozdat vysoký damage, ale také udělat chybu.
- **Snövit-Blackhand** - Podle Raider.io má skóre 1,375. Jedná se o průměr až lehký nadprůměr. Jeho profil nesvědčí o výrazných úspěších. V záznamech v našem systému má pouze jeden dungeon a to Spires of Ascension 16, což už je klíč vyšší obtížnosti. Dle našeho systému vyniká hlavně v DamageDoneScore a ve FailureDamageScore, DamageReceivedScore, DeathsScore a v DamageDoneScore. Podle těchto čísel si zaslouží přední příčky.
- **Älde-Uldum** - Podle Raider.io a celkově jeho profilu se jedná o nadprůměrného hráče. Dle Raider.io má skóre 1,404. V našem systému má jeden záznam. Jedná se o Sanguine Depths 14, což je běžná obtížnost klíče pro hráče těchto kvalit. Dle záznamu má běh dungeonem téměř bez chyby a skóre mu navíc zvedá HealingScore, které by se u damage dealera nemělo moc počítat a možná právě proto se lehce nadprůměrný hráč dostal mezi nejvyšší ranky.
- **Snuwut-Draenor** - Podle jeho profilu se jedná o vedlejší postavu velice kvalitního hráče módu Mythic+. Na své hlavní postavě má Raider.io skóre 1,710, což je velice dobré. V našem systému má jeden záznam a jedná se o Mists of Tirna Scitheh 13, což je pro hráče této úrovně nízký klíč. Navíc měl na tomto runu kvalitní tým, pro který byl tento klíč spíše jednoduchý. V tomto případě je snadné vidět, proč získal v našem systému tak vysoké skóre.
- **Rastafüry-Archimonde** - Jedná se o vedlejší postavu velice dobrého hráče, který má na své hlavní postavě Raider.io skóre 1,990. Vysoké skóre v našem systému si vysloužil dungeonem Mists of Tirna Scitheh 13 ve stejném runu jako hráč Snuwut-Draenor, což svědčí o čistém a bezchybném runu. Takové hráče bych přesně čekal na předních příčkách našeho systému.
- **Flappydab-Blackrock** - Jedná se o vedlejší postavu velice dobrého hráče módu Mythic+ i jiných PvE herních módů. Na své hlavní postavě má Raider.io skóre 1,945 a je to tak nejlépe hodnocený hráč dle Raider.io v našem systému. Proto dává smysl, že získal tak vysoké skóre. Toto skóre získal ve stejném dungeonu jako Sneakyball-Thrall - Necrotic Wake 12, což svědčí o hladkém, takřka bezchybném průchodu dungeonem.

Nejhorších 10 skóre v našem systému obsadilo těchto 10 hráčů.

- **Bigbadbull-Saurfang** - Jedná se o hráče s poměrně špatným vybavením. Nehraje zas tak často a chodí převážně nízké klíče. Kvůli svého vybavení nepůsobí nijak veliké poškození. Podle jeho jediného záznamu, který se mu vůbec nepovedl, se moc nedívá

co se kolem něj děje a nechává se trefovat veškerými mechanikami. Dle Raider.io má skóre 660, což je hodně podprůměrné.

- **Grönflicka-Defias Brotherhood** - V době pořízení záznamu tohoto hráče měl na své postavě prakticky nepoužitelné vybavení. V záznamu dungeonu se nechal boostovat skupinou silných hráčů. Na Raider.io má skóre 501, což je opravdu nízké. Ačkoliv šel nízkou úroveň klíče, kde vás chyby prakticky nic nestojí a zároveň byl ve skupině s opravdu silným healerem, podařilo se mu 4x umřít. Podle záznamu se nechával trefovat skoro každou mechanikou. Jeho damage done není sice vůbec špatný vůči jeho vybavení, ale to může být i důvod, proč nedokázal dodržovat herní mechaniky.
- **Keriggen-Drak'thul** - Tento hráč má dle Raider.io skóre 332. V některých dungeonech nikdy ani nebyl. Je to hráč, který mód Mythic+ moc nehraje a tomu odpovídá i jeho výkon. Není divu, že skončil na spodních příčkách našeho systému.
- **Warcore-TwistingNether** - Jedná se o alta (vedlejší postavu - alt jako alternative) velice zkušeného a kvalitního hráče, který hraje velice často. Na hlavní postavě má dle Raider.io skóre 1,905. Tento dungeon se mu podle všeho ale vůbec nepovedl. Jedná se o hráče, který hraje jenom damage dealery a má vybavení pouze pro damage dealery. Na své vedlejší postavě však zkoušel tankovat na klíči úrovni 11, což není mnoho, ale zároveň chyby tam už jsou poměrně přísně trestány. Role tanka je velice odlišná od role damage dealera a jsou tam vyžadovány úplně jiné herní dovednosti. V jeho skupině se také vyskytovali jiní hráči, kteří jsou také v tomto seznamu nejhorších hráčů a i to mohlo ovlivnit jeho výkon. Nicméně si nemyslím, že by měl v našem systému být do budoucna hodnocen takhle nízko.
- **Elsje-Doomhammer** - Jedná se o čistě průměrného hráče, který má na Raider.io skóre 1299. U tohoto hráče jsme našli veliký nedostatek v přerušování nebezpečných kouzel, což když se zkombinuje s jedním z vašich nejhorších výkonů, tak to potom může vést až takto nízkému ratingu v našem systému. Spíše si ale myslím, že si nezasouží mít takhle nízké skóre a ve svých příštích záznamech by si jej dozajista spravil.
- **Skhizm-BurningSteppes** - Tento hráč dosáhl 716 Raider.io skóre. Má sice dokončené všechny dungeony, ale některé z nich nikdy nedokončil včas a to je opravdu velice slabé. To, že je tento hráč hodnocen takhle nízko v našem systému, dává určitě smysl.
- **Najaha-TwistingNether** - V tomto případě se jedná o podprůměrného hráče. Dungeony chodí pravidelně a na webu Raider.io má skóre 1,132, které už je výrazně podprůměrné. Vzhledem ke kvantitě jeho odchozených dungeonů a takto nízkému skóre mi dává smysl, že skončil v našem systému takhle nízko.
- **Øneshøt-Kazzak** - Tento hráč dosáhl v systému Raider.io skóre 908, což svědčí o tom, že není moc kvalitní hráč. Zároveň u něj můžeme pozorovat, že jeho nejlepší dungeon runy jsou na o dost vyšších klíčích, než chodí normálně, takže můžeme říct, že se mohl nechat vyboostit.
- **Hollyragnar-Kazzak** - Jedná se o hráče, který pravidelně chodí nízké úrovně dungeonů a pravidelně se nedostává do úrovní vyšších i přes to, že hraje často. Na webu Raider.io má skóre 640, což je opravdu nízké. To, že je v našem systému taky hodnocen nízko, dává smysl.
- **Slagschaduw-Sunstrider** - V tomto případě musel tento hráč mít opravdu špatný den. Nejedná se vůbec o špatného hráče, dlouhodobě podává solidní výkony. Na webu Raider.io má skóre 1480, což není málo. V našem systému by měl být rozhodně po více zaznamenaných dungeonech hodnocen lépe.

Až na výjimky si náš ratingový systém vede poměrně dobře. V top deseti hráčích se až na jedinou výjimku vyskytují zkušení a kvalitní hráči. V desítce nejhorších hráčů se pak vyskytují převážně nezkušení a špatní hráči. Různé výjimky jsou dány nedostatkem záznamů, kdy právě mezi nejlepší hráče mohou proniknout průměrní, kterým se jejich jediný záznam právě povedl, a to stejné platí i pro nejhorší hráče. V porovnání se systémem Raider.io si to však nevede zle, protože pokud by hráč měl jeden záznam v systému Raider.io, měl by poté velice nízké skóre. Nejvyšší světový klíč zvládnutý včas je úroveň 27 a za její včasné dokončení vám systém Raider.io vyměří 509 bodů, což je dnes v Raider.io systému hodně nízké skóre.

7.3.2 Predikce

S týmem známých spoluhráčů jsme odchodili dungeony od úrovně 7 do úrovně 18 a nechali si pokaždé predikovat úspěch našim systémem. V tabulce predikcí je úspěch v dungeonu značen 1 a neúspěch v dungeonu značen -1. Predikce úspěchu známých spoluhráčů našim systémem je zobrazena v tabulce 7.1.

Úroveň	7	8	9	10	11	12	13	14	15	16	17	18
Predikce	1	1	1	1	1	1	1	1	1	1	-1	-1
Realita	1	1	1	1	1	1	1	1	1	1	-1	-1

Tabulka 7.1. Predikce úspěchu naší skupiny hráčů v dungeonech úrovní 7-18.

Dále jsem složil skupinu z top hráčů v našem systému a nechal jsem pro ně vygenerovat predikci na úrovních 15 až 26. Jelikož však tyto hráče neznám a nemám na ně kontakt, nemohli jsme predikci ověřit. Nicméně dle jejich herních profilů a záznamů odchozených dungeonů a Raider.io skóre jsem schopen odhadnout, jestli predikce dává smysl. Predikce nejlepších hráčů je zobrazena v tabulce 7.2.

Úroveň	15	16	17	18	19	20	21	22	23	24	25	26
Predikce	1	1	1	1	1	1	1	1	1	1	1	1

Tabulka 7.2. Predikce nejlepší možné skupiny hráčů zaznamenaných v našem systému v dungeonech úrovní 15-26.

Nakonec jsem vzal skupinu nejhorších hráčů u nás v systému a udělal predikci i pro ně na úrovních 2-13, zobrazenou tabulkou 7.3. Jelikož však tyto hráče neznám a nemám na ně kontakt, nemohli jsme predikci ověřit. Nicméně dle jejich herních profilů a záznamů odchozených dungeonů a Raider.io skóre jsem schopen odhadnout, jestli predikce dává smysl. Jedná se o hráče Hollyragnar-Kazzak, Najaha-TwistingNether, Bigbadbull-Saurfang, Keriggen-Drak'thul a Skhizm-BurningSteppes.

Úroveň	2	3	4	5	6	7	8	9	10	11	12	13
Predikce	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1

Tabulka 7.3. Predikce nejhorší možné skupiny hráčů zaznamenaných v našem systému v dungeonech úrovní 2-13.

V případě skupiny známých spoluhráčů došlo ke stoprocentní přesnosti predikce. Tohle může být způsobeno přefitováním modelu na naši skupinu, protože právě my jsme sbírali data nejvíce. Z celkového datasetu 65 dungeon runů jsem já osobně šel 16x

na své hlavní postavě, která byla použita tady v tomto testu. Na druhou stranu po tolika dungeon runech už se dá pěkně odhadnout průměrný průběh dungeonem hráče na dané postavě. Také predikce nemusela být stoprocentní, protože naše úspěšnost v minulosti na úrovni dungeonu 17 je tak 1:2 - tudíž jeden úspěch na dva neúspěchy, takže došlo by-li v tomto případě k úspěchu, predikce by už nebyla bezchybná.

V případě skupiny nejlepších hráčů systému je predikce příliš optimistická. Podle Raider.io se jedná o hráče se skóre pohybujícím se od 1500 do 2000. To odpovídá dungeonům zhruba úrovně 20. Světově nejvyšším klíčem dokončeným včas je ke dni 17.5.21 v této sezóně je 27. [3]. Z toho lze vyvodit, že úrovně 25 a 26 jsou určitě mimo dosah této skupiny. Toto je způsobené tím, že tito hráči mají v systému jenom jeden záznam, a to na klíči jim pohodlné a nenáročné úrovně, kde jsou schopni zahrát téměř bezchybný výsledek, a také nedostatek dat o rekordně vysokých klíčích, protože k těmto rekordním klíčům a lidem, kteří se je pokouší zdotat, zkrátka nemáme přístup. Systém si poté myslí, že nejlépe hodnocení hráči v našem systému zvládnou úplně vše.

V případě nejhorších hráčů nastává velice podobná situace. Predikce je zde příliš pesimistická, protože ačkoliv je nejedná o vysoce kvalitní hráče, úrovně klíče 2 a 3 by měli určitě zvládnout, dle jejich vybavení. Na tyto úrovně mají tito hráči extrémně dobré vybavení a jakákoliv chyba by je neměla být schopna zabít. Zároveň by měli ve svém vybavení působit takové poškození, že by neměli dostat prostor udělat žádnou chybu. Tato predikce vyšla špatně z důvodu nedostatku dat o dungeonech nižší úrovně a hlavně každý z těchto hráčů má v systému pouze jeden záznam na klíči vyšším, než je jejich komfortní zóna, a tím pádem se může stát, že udělají naprosto špatný výsledek. Když tento dungeon navíc nedokončí včas, nemají poté žádný záznam, který by stihli včas, což má na predikci určitě vliv.

Kapitola 8

Instalace

8.1 Instalace addonu

Addon nemá žádné instalační skripty a ani jej není třeba nějak kompilovat či instalovat. Zdrojový kód addonu stačí umístit do správné složky v adresáři s hrou World of Warcraft a hra si potom sama kód najde, přeloží a spustí.

Nacházíme-li se v kořenovém adresáři hry, pak extrahujeme složku s kódy do složky `./_retail_/Interface/AddOns/`. Po příštím přihlášení do hry by měl být addon aktivní a připravený ke sběru dat. Můžeme se o tom přesvědčit v sekci výběru postav před nalogováním do hry. Vlevo dole je tlačítko AddOns a po kliknutí na něj se objeví seznam, kde by měl být addon zobrazen jako KeystonePro a jeho select box by měl být zaškrtnutý. Dále je doporučeno mít zaškrtnutý také checkbox v pravém horním rohu okna se seznamem addonů Load out of date AddOns.



Obrázek 8.1. AddOn interface přímo ve hře [4]

Na obrázku 8.1 můžeme vidět AddOn interface přímo ve hře. Lze v něm zapínat a vypínat načtené addony z adresáře s hrou.

8.2 Instalace uploaderu

Uploader nemusí být instalován, stačí si rozbalit složku KeyStonePro Uploader ve kterém se nachází skript pro OS Windows, nebo pro MacOS. V kořenovém adresáři složky KeyStonePro Uploader najdeme skripty KeyStonePro Uploader.bat pro Windows nebo

ve složce bin/KeyStonePro Uploader pro MacOS. Pro správné fungování uploadu známů na server musíme být odhlášeni ze své postavy ve hře.

Kapitola 9

Závěr

V této práci jsme uvedli co je to mód Mythic+ ve hře World of Warcraft a jakými do-
savadními způsoby se v něm dají individuálně hodnotit hráči. Dále jsme nabídli lehký
vhled, jak se vytváří addony do různých her. Navrhli jsme vlastní systém hodnocení
hráčů v herním módu Mythic+ hry World of Warcraft. Dále jsme nastínili možné techn-
logie pro implementaci tohoto systému a pro každou část systému jednu z nich zvolili.
Z daných částí systému jsem implementoval a představil implementaci herního addonu
do hry World of Warcraft, který sbírá data o hráčích a jejich akcích v průběhu herního
módu Mythic+, a také veřejně přístupný frontend systému v podobě webové aplikace,
sloužící k vizualizaci a zobrazení dat uživatelům a zájemcům o systém a také nabízející
predikce úspěchu jednotlivých týmů hrajících tento herní mód. Pomocí addonu jsme
sesbírali data, která jsme odeslali na server systému a tam jsme je zpracovali a uložili.
Tato data jsem analyzoval a zjistil, které atributy data nejlépe separují a mají vliv na
úspěch či neúspěch skupiny. Zjistil jsem, že největší vliv na úspěch mají tři z těchto osmi
atributů. Z těchto dat jsem poté vytvořil dataset použitý k natrénování klasifikátoru,
predikujícího úspěch skupiny v dungeonu dané úrovně. Natrénovat klasifikátor tak, aby
dával reálné výsledky se nám spíše nepodařilo, a to kvůli celkovému nedostatku dat,
nízké reprezentaci extrémně nízkých a extrémně vysokých úrovní v datasetu a nedo-
statku informací o cizích hráčích, kdy většina hráčů má v systému pouze jediný záznam,
který nemusí odpovídat jejich obvyklému výkonu, zvláště pak hráči s nejlepším a nej-
horším skórem. Dále jsme celý systém otestovali, uvedli do chodu a poměrně úspěšně
porovnali s jinými ratingovými systémy, kdy hodnocení hráčů dává smysl - horší výkony
jsou hodnoceny hůře než ty lepší. Tímto jsme splnili všechny cíle této práce.

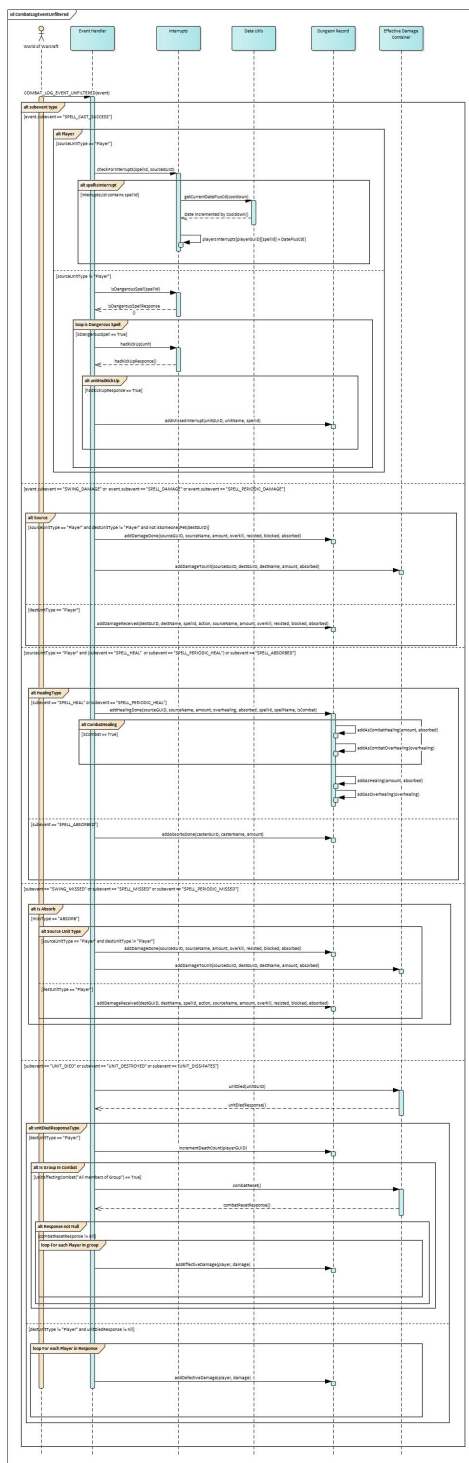
Kapitola 10

Future Work

V rámci future work bych se chtěl hlavně zaměřit na vyzkoušení jiných datových datových modelů s jiným vzorcem pro výpočet jednotlivých featur a taky přidání featur úplně nových. Chtěl bych rozšířit addon, aby sbíral další atributy - například overall dps, to je damage done za sekundu v combatu. To znamená oddělení damage od pohybu v dungeonu. Také bych chtěl nasbírat mnohem větší dataset, abych poté mohl predikci lépe optimalizovat a použít například metody a klasifikátory fungující na velkém objemu dat. Dále bych chtěl rozšířit frontend aby zobrazoval například leaderboard hráčů zaznamenaných v systému, přidat rozkliknutelný detail jednotlivých hráčů zobrazených po uživatelském requestu a také je třeba zlepšit vizuální stránku frontendu celkově a nabídnout denní a noční mód. Systém by také mohl zobrazovat v jakém aspektu by se jednotliví hráči mohli zlepšit a naopak ve kterém aspektu vynikají. Například máme hráče který dává nadprůměrný damage oproti lidem hrající podobnou postavu, ale zůstává v počtu interruptů a dobře si je nehlídá. Dále bych chtěl zablokovat hráče s nízkým počtem záznamů v systému pro predikci, protože může jít o extrémní případy - extrémně vydařený a extrémně nepodařený výsledek, který poté neodpovídá jejich standartnímu výkonu.

Příloha A

Přílohy



Obrázek A.1. Sekvenční diagram eventu COMBAT_LOG_EVENT_UNFILTERED

Literatura

- [1] Blizzard Entertainment. *World of Warcraft Home Page*. 2021.
<https://worldofwarcraft.com/en-us/>.
- [2] *Blizzard website*. 2021.
<https://www.blizzard.com/en-us/>.
- [3] *Raider.io - World of Warcraft Rankings for Mythic+ and Raid Progress*. 2021.
<https://raider.io/>.
- [4] Blizzard Entertainment. *World of Warcraft, the game*. 2021.
<https://worldofwarcraft.com/en-us/>.
- [5] *Raider.io*. 2021.
<https://raider.io/characters/eu/drakthul/Dridek>.
- [6] LEWIS LEWIN. *Best GTA 5 Star Wars Mods Every Fan Should Try*. 2021.
<https://www.fandomspot.com/gta5-star-wars-mods/>.
- [7] *Minecraft Forge downloads*. 2021.
<https://files.minecraftforge.net/net/minecraftforge/forge/>.
- [8] DAVID DODGE. *How to Install Minecraft Forge: The Ultimate Guide [Updated 2021]*. 2020.
<https://codakid.com/minecraft-forge/>.
- [9] Anastasia Maillot. *The 10 Weirdest Skyrim Mods Ever*. 2021.
<https://gamerant.com/weirdest-skyrim-mods/>.
- [10] Martin Lukeš. 2021. Praha, 2021. Bakalářská práce. ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE, Fakulta elektrotechnická. Vedoucí práce Ing. Jiří Šebek..
- [11] Xrystal. *WoWInterface*. 2021.
<https://www.wowinterface.com/forums/showthread.php?p=338485#post338485>.
- [12] Jason Brownlee. *Perceptron Algorithm for Classification in Python*. 2020.
<https://machinelearningmastery.com/perceptron-algorithm-for-classification-in-python/>.
- [13] Sushanth SREENIVASA. *Radial Basis Function (RBF) Kernel: The Go-To Kernel*. 2020.
<https://towardsdatascience.com/radial-basis-function-rbf-kernel-the-go-to-kernel-acf0d22c798a>.
- [14] *Sklearn.ensemble.AdaBoostClassifier*. 2020.
<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html>.
- [15] Akash Desarda. *Understanding AdaBoost*. 2019.
<https://towardsdatascience.com/understanding-adaboost-2f94f22d5bfe>.
- [16] Squishei. *Everything About Mythic+ Keystones and Dungeons*. 2021.
<https://www.wowhead.com/mythic-keystones-and-dungeons-guide>.

-
- [17] Alexandre Gonfalonieri. *Dealing with the Lack of Data in Machine Learning*. 2019.
<https://medium.com/predict/dealing-with-the-lack-of-data-in-machine-learning-725f2abd2b92>.
- [18] Nitesh V. Chawla, Kevin W. Bowyer, USA Lawrence O. Hall a W. Philip Kegelmeyer. *SMOTE: Synthetic Minority Over-sampling Technique*. 2002.
<https://www.jair.org/index.php/jair/article/view/10302>.
- [19] About. *The programming language Lua*. 2020.
<https://www.lua.org/about.html>.
- [20] Google. 2021.
<https://angular.io/guide/what-is-angular>.
- [21] Daniel Fleury. *Who Uses Angular in 2021? 12 Global Websites Built With Angular*. 2021.
<https://trio.dev/blog/companies-use-angular>.
- [22] Facebook. *React – A JavaScript library for building user interfaces*.
<https://reactjs.org/>.
- [23] Evan You. 2021.
<https://vuejs.org/>.
- [24] Shaumik Daityari. *Angular vs React vs Vue: Which Framework to Choose in 2021*. 2021.
<https://www.codeinwp.com/blog/angular-vs-vue-vs-react/>.
- [25] Luca Spezzano. *Google, Apple and Other Users of Vue.js*. 2020.
<https://medium.com/notonlycss/google-apple-and-other-users-of-vue-js-e4505359e5d5>.
- [26] *Stack Overflow Developer Survey 2018*. 2018.
<https://insights.stackoverflow.com/survey/2018/#most-loved-dreaded-and-wanted>.
- [27] Md Kamaruzzaman. *Top 10 In-Demand Web Development Frameworks in 2021*. 2020.
<https://towardsdatascience.com/top-10-in-demand-web-development-frameworks-in-2021-8a5b668be0d6>.
- [28] John Potter. *vue vs @angular/core vs react*. 2021.
<https://www.npmtrends.com/vue-vs-@angular/core-vs-react>.
- [29] Numpy. *NumPy Home Page*. 2021.
<https://numpy.org/>.
- [30] David Cournapeau. *Scikit-learn Home Page*. 2021.
<https://scikit-learn.org/stable/>.
- [31] Adam Paszke, Sam Gross, Soumith Chintala a Gregory Chanan. *PyTorch Home Page*. 2021.
<https://pytorch.org/>.
- [32] Springboard India. *Best language for Machine Learning: SpringboardIN Blog*. 2020.
<https://in.springboard.com/blog/best-language-for-machine-learning/>.
- [33] Priyamvada PS. *5 Best programming languages for Machine Learning in 2021*. 2021.
<https://content.techgig.com/5-best-programming-languages-for-machine-learning-in-2021/articleshow/81782213.cms>.

- [34] Konduit. *Deep Learning for Java Home Page*. 2021.
<https://deeplearning4j.org/>.
- [35] The Apache Software Foundation. *Mllib: Apache Spark*. 2021.
<https://spark.apache.org/mllib/>.
- [36] Scott Chacon a Jason Long. 2021.
<https://git-scm.com/>.
- [37] *GitHub commands 21.03% market share in Software Configuration Management*. 2021.
<https://enlyft.com/tech/products/github>.
- [38] *COMBAT LOG EVENT*. 2021.
https://wowpedia.fandom.com/wiki/COMBAT_LOG_EVENT.
- [39] Google. 2021.
<https://material.angular.io/>.
- [40] BLIZZARD ENTERTAINMENT INC. ALL RIGHTS RESERVED. 2021.
<https://develop.battle.net/>.
- [41] Jaliborc. *Jaliborc/WoWUnit*. 2020.
<https://github.com/Jaliborc/WoWUnit>.
- [42] Gregg Van Hove. *jasmine/jasmine*. 2021.
<https://github.com/jasmine/jasmine#installation>.
- [43] Friedel Ziegelmayer. *How It Works*. 2021.
<https://karma-runner.github.io/6.3/intro/how-it-works.html>.
- [44] Google. 2021.
<https://angular.io/guide/testing>.