



TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií ■

Softwarový robot pro Discord server

Ročníkový projekt

Studijní program:

B2646 – Informační technologie

Studijní obor:

1802R007 – Informační technologie

Autor práce:

Tomáš Novotný

Vedoucí práce:

Ing. Lukáš Matějů, Ph.D.

Abstrakt

Tato práce se zabývá vytvořením softwarového robota pro komunikační platformu Discord. Aplikace je vytvořena na platformě .NET pomocí knihovny pro tvorbu botů, DSharpPlus. Bot je primárně nastaven pro práci s mobilní hrou Pokémon Go, která je velice populární po celém světě a využívají ji desítky milionů hráčů. Ve hře je nutná spolupráce v týmu založená na vzájemné domluvě mezi hráči. Tento bot by měl usnadnit a zefektivnit komunikaci mezi nimi, také by měl pomoci novým hráčům v pochopení systému komunikace na Discord serveru. Bot reaguje na příkazy uživatele a zároveň je mu schopen zobrazovat přehledná data o hře, která jinak nejsou veřejně dostupná.

Klíčová slova: bot, Pokémon GO, Discord, .NET, DSharpPlus

Abstract

This thesis deals with the creation of a software robot for the Discord communication platform. The application is built on the .NET platform using the bot creation library, DSharpPlus. The bot is primarily set up to work with the mobile game Pokémon Go, which is very popular worldwide and used by tens of millions of players. The game requires teamwork based on mutual cooperation between players. This bot should make communication between them easier and more efficient, also helping new players to navigate and understanding rules of communication on Discord server. The bot responds to the user's commands and is also able to display clear game data that is otherwise not publicly available.

Key words: bot, Pokémon GO, Discord, .NET, DSharpPlus

Prohlášení

Prohlašuji, že svou práci k ročníkovému projektu jsem vypracoval samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce. Jsem si vědom toho, že se na tuto práci plně vztahuje zákon č. 121/2000 Sb. o právu autorském, zejména § 60 – školní dílo. Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé práce pro vnitřní potřebu Technické univerzity v Liberci. Užiji-li tuto práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše. Jsem si vědom následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

V Liberci dne 4. září 2022

Podpis:

Poděkování

Tímto bych chtěl poděkovat Ing. Lukáši Matějů, Ph.D., za spolupráci a konzultace při zpracování práce k ročníkovému projektu.

Obsah

| | |
|--|----|
| Úvod..... | 8 |
| 1 Použité technologie..... | 9 |
| 1.1 Microsoft Visual Studio | 9 |
| 1.2 .NET | 9 |
| 1.3 Entity Framework Core..... | 9 |
| 1.4 DSharpPlus | 10 |
| 1.5 SQLite | 10 |
| 1.6 Discord..... | 10 |
| 1.7 DB Browser for SQLite | 11 |
| 2 Pokémon GO | 13 |
| 2.1 Ingress | 13 |
| 2.2 Motivace | 14 |
| 2.3 Discord server | 14 |
| 2.4 Textové kanály | 14 |
| 2.5 Systém hlášení raidů | 15 |
| 2.6 Bot a jeho funkce | 16 |
| 2.7 Možné rozšíření | 16 |
| 3 Technická pravidla..... | 17 |
| 3.1 S2 buňky | 17 |
| 3.2 Pravidla zobrazení bodů ve hře | 17 |
| 3.3 Získání dat o bodech | 18 |
| 3.4 Zpracování dat o bodech..... | 19 |
| 3.5 Vznik konfliktů | 19 |
| 4 Implementace aplikace | 20 |
| 4.1 Databáze..... | 20 |
| 4.2 Souborová struktura | 20 |
| 4.3 Příkazy bota | 21 |
| 4.3.1 Nahrávání dat | 22 |
| 4.3.2 Kontrola dat..... | 24 |
| 4.3.3 Změna názvu bodu | 24 |
| 4.3.4 Změna typu bodu | 25 |

| | | |
|----------------------|-----------------------|----|
| 4.3.5 | Procházení bodů | 25 |
| 4.3.6 | Hlášení raidů | 26 |
| 4.3.7 | Ukázání zájmu..... | 27 |
| 4.3.8 | Nastavení srazu | 27 |
| Závěr..... | | 29 |
| Literatura | | 30 |
| Seznam obrázků | | 31 |

Seznam zkratek

| | |
|------|--|
| API | Application Programming Interface |
| PvP | Player vs Player (hráč proti hráči) |
| PvE | Player Versus Environment (hráč proti hře) |
| IITC | Ingress Inter Total Conversion |
| txt | Text |
| csv | Comma-separated values (hodnoty oddělené čárkami) |
| json | JavaScript Object Notation (JavaScriptový objektový zápis) |

Úvod

Tento ročníkový projekt se zabývá vytvořením softwarového robota (dále bot) pro komunikační platformu Discord. Bot má za úkol usnadnit doposud problematickou komunikaci mezi uživateli pro mobilní hru Pokémon GO. Jeho úkolem je vytvořit a udržovat databázi bodů, které se ve hře nacházejí, a zpřístupnit uživatelům tato data, která budou jednoduše a přehledně zobrazena. Cílem tohoto projektu je usnadnit a zefektivnit domluvu mezi jednotlivými uživateli hry a podpořit týmovou spolupráci hráčů. Zároveň by měl projekt pomoci novým uživatelům zorientovat se v nastaveném systému komunikace na Discord serveru, jelikož rychlá, neorganizovaná komunikace s ostatními uživateli pro ně může být komplikovaná.

První část práce je zaměřena na představení jednotlivých technologií, které byly při tvorbě bota použity a umožňují jeho fungování.

Dále je pozornost věnována samotné hře Pokémon GO, jejímu provozu v praxi a vysvětlení základních pojmů. Je zde představena autorova motivace k vytvoření tohoto projektu a přínos, který výsledný bot bude mít. Zároveň autor práce nastiňuje, jakým způsobem bude bot v prostředí platformy Discord pracovat a jaké má funkce.

Ve třetí části práce najdeme technická pravidla, dle kterých se bot řídí. Jsou zde vysvětleny všechny důležité pojmy a postupy, které jsou důležité pro tuto práci.

Závěrečná část se zaměřuje na implementaci samotného bota, popis funkcí a názornou ukázkou v praxi. Popisuje také strukturu databáze a programu. Jsou zde vysvětleny postupy, při zpracování dat.

1 Použité technologie

1.1 Microsoft Visual Studio

Microsoft Visual Studio (dále Visual Studio) je vývojové prostředí vytvořené společností Microsoft. Využívá se pro vývoj konzolových, webových, mobilních aplikací nebo webových stránek. Nejnovější verze vydání Visual Studio 2022 obsahuje vylepšenou verzi IntelliSense, kterou je výkonná sada nástrojů IntelliCode. Ta na základě již napsaného kódu, strojového učení a získané znalosti z půl miliónu kódových vzorů dokáže vyhodnotit, co bude chtít programátor napsat a vyplnit na něj celý řádek, případně celý cyklus. Visual Studio integruje možnost používat Git pro klonování kódu na cloudové úložiště. Díky chytrému zabudovanému ladění lze kód jednoduše procházet a odhalit chyby. Existují 3 verze: Community pro studenty a menší vývojáře, Professional a Enterprise pro náročnější projekty.¹

1.2 .NET

.NET je bezplatná, open source vývojová platforma, která je určena pro vytváření různých aplikací. Jedná se o spojení programovacích jazyků a knihoven a CLR (Common Language Runtime), což je běhové prostředí, které zajišťuje běh a kompilaci aplikací. .NET podporuje programovací jazyky C# (zvolen pro tuto práci), F# a Visual Basic. C# je jednoduchý, moderní, objektově orientovaný a typově bezpečný jazyk. Pomocí .NETu lze vyvíjet aplikace pro web, osobní počítače, mobilní telefony, hry či cloud. Jde se o multiplatformní technologii, s kterou můžeme stejný kód spouštět na systému Windows, macOS, Linux, Android nebo iOS. Aplikace využívá nejnovější verzi .NET 6.0.²³

1.3 Entity Framework Core

Entity Framework Core (dále EF Core) zajišťuje přenos dat mezi aplikací s databází. Je součástí .NET a umožňuje nám pracovat s entitami a atributy jako s třídami a vlastnostmi. Usnadňuje sledování změn, zavoláním metody SaveChanges se provede automatická aktualizace přidáných, změněných či odebraných objektů a následně tyto změny promítne v databázi. Po definování tříd entit můžeme vytvořit migraci, ze které se vygeneruje počáteční

¹ Microsoft [online]. [cit. 2022-08-20]. Dostupné z: <https://visualstudio.microsoft.com/cs/vs/>

² HERCEG, Tomáš. Úvod do .net frameworku. In: *Dotnetportal.cz* [online]. 3. 4. 2009 [cit. 2022-08-20]. Dostupné z: <https://www.dotnetportal.cz/clanek/125/Uvod-do-NET-Frameworku>

³ Microsoft [online]. [cit. 2022-08-20]. Dostupné z: <https://dotnet.microsoft.com/en-us/learn/dotnet/what-is-dotnet>

sada tabulek.⁴ EF Core pracuje s dotazovacím jazykem LINQ (Language-Integrated Query), který je kompatibilní s různými typy zdrojů a formáty dat. Programátor tedy pracuje se stejnými objekty bez ohledu na cílovou databázi.⁵

1.4 DSharpPlus

DSharpPlus je neoficiální multiplatformní .NET knihovna, tzv. „wrapper“, založený ze zrušeného projektu DiscordSharp, který se používá k propojení s oficiálním Discord API. Je přepsaná tak, aby odpovídala standardům API, a zaměřuje se na širší škálu implementace .NETu. Ověření je zajištěno pomocí soukromého klíče (token), který pro přihlášení vygeneruje Discord API a je uložen na straně klienta. Aplikace využívá verzi 3.2.3.⁶

1.5 SQLite

SQLite je nejpoužívanější knihovna nástrojů, databázový systém bez použití serveru nebo služby. Jednotlivé databáze se ukládají jako soubory. Výhodou je malá velikost souborů (jednotky MB), rychlé načítání, žádná nebo jen minimální konfigurace, snadná přenositelnost a multiplatformní použití. SQLite je transakční, pokud se tedy transakci nepovede úspěšně dokončit např. pádem programu nebo operačního systému, tak se neprovede a databáze se vrátí do původního validního stavu. Přístup k databázi je umožněn z více instancí. Nevýhodou je naopak nemožnost čtení a zapisování zároveň, protože při ukládání informací se databáze uzamkne. Postrádá některé datové typy, které lze nahradit základními (Bool → Int nebo DateTime → String). Není možná správa oprávnění a uživatelů na úrovni databáze, ale je možné zabezpečit databázový soubor pomocí operačního systému. Podle vývojářů je záměr podporovat SQLite do roku 2050.⁷

1.6 Discord

Komunikační platforma Discord je bezplatná aplikace pro hlasovou, video a textovou komunikaci. Probíhá na jednotlivých serverech, které jsou většinou zaměřeny na jedno téma nebo zastřešují určitou komunitu lidí, např. fanoušky hry *Pokémon GO* nebo známé hudební skupiny apod. Přístupnost serveru může být veřejná nebo soukromá. Pouze registrovaní

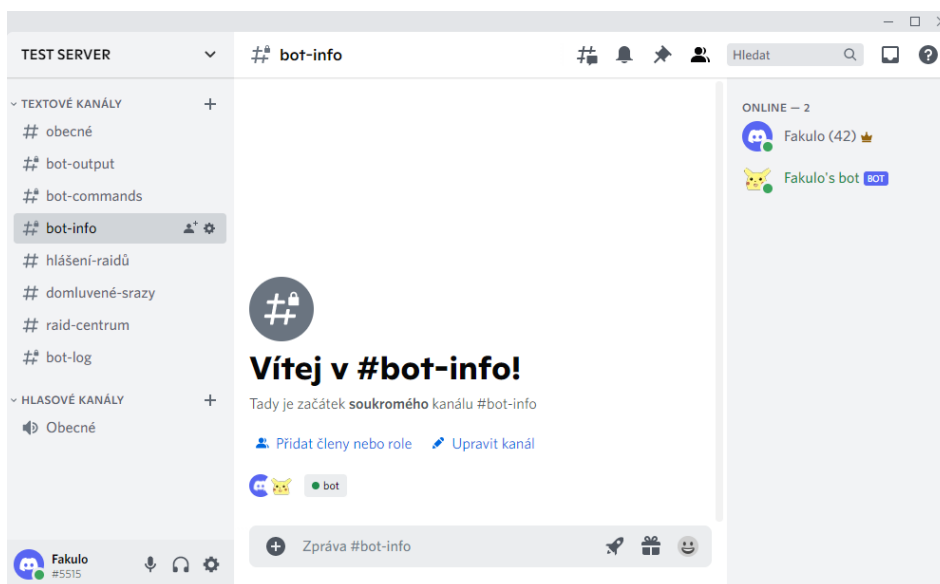
⁴ Microsoft [online]. [cit. 2022-08-20]. Dostupné z: <https://docs.microsoft.com/cs-cz/ef/core/>

⁵ Microsoft [online]. [cit. 2022-08-20]. Dostupné z: <https://docs.microsoft.com/cs-cz/dotnet/csharp/linq/>

⁶ DSharpPlus Contributors [online]. [cit. 2022-08-21]. Dostupné z: <https://dsharpplus.github.io/>

⁷ SQLite [online]. [cit. 2022-08-20]. Dostupné z: <https://www.sqlite.org/about.html>

uživatelé se mohou připojit, a to na několik serverů najednou. Výměna informací probíhá v hlasových a textových kanálech. Uživatelé mohou dostat určité role, které specifikují, jaké budou mít pravomoci na serveru. Jde např. o omezený přístup k jednotlivým hlasovým a textovým kanálům, využívání kanálu pouze pro čtení nebo možnost zablokování ostatních uživatelů.⁸



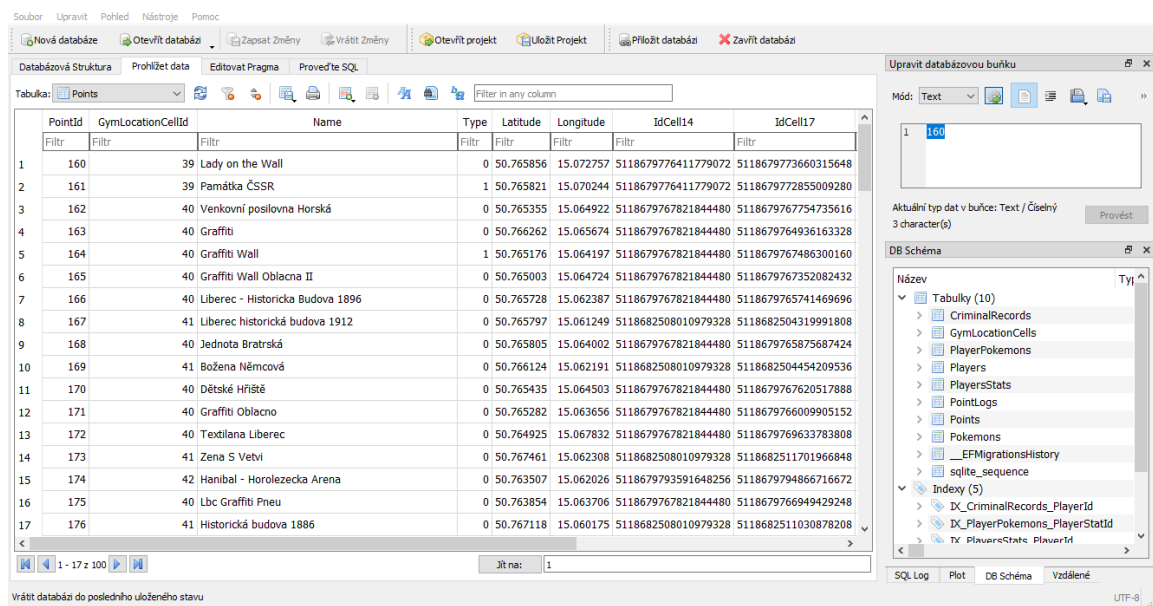
Obrázek 1 Grafické rozhraní testovacího serveru na Discordu

1.7 DB Browser for SQLite

DB Browser for SQLite (dále DB Browser) je open source nástroj pro vytváření, navrhování, čtení a editaci databázových souborů kompatibilních s SQLite. S jeho pomocí můžeme vyhledávat uložené záznamy a kontrolovat jejich správné zapsání do souboru. Software zároveň umožňuje jednoduchý import a export tabulek do souborů formátu txt nebo csv. Aplikace je intuitivní a přehledná.⁹

⁸ Discord [online]. [cit. 2022-08-20]. Dostupné z: <https://discord.com/safety/360044149331-What-is-Discord>

⁹ DB Browser for SQLite [online]. [cit. 2022-08-20]. Dostupné z: <https://sqlitebrowser.org/about/>



Obrázek 2 Grafické rozhraní programu DB Browser

2 Pokémon GO

Pokémon GO je populární mobilní hra, jež využívá původní japonský námět „kapesních příšerek“ pokémonů z 90. let a navazuje na tradici původních konzolových a karetních her, televizního seriálu apod. Jejím cílem je nasbírat všechny pokémony, kteří existují ve virtuálním světě hry, ale zároveň se skrze mobilní telefon dostávají do reality.

Pokémoni se objevují po celé mapě reálného světa, nejčastěji v parcích, na pěších cestách nebo u pokéstopů a gymů (místa reflektující zajímavé objekty v realitě, např. památník, dětské hřiště, historické či neobvyklé stavby, zdobené výklenky atd.), kde hráči sbírají pokébally (míčky na chytání) a berries (krmivo pro vylepšení chytání). Své pokémony si hráč může za pomoci nasbíraného stardustu a candies (druh měny) vyvíjet a vylepšovat. Obě možnosti hráč získává za chytání pokémonů, candies i za prodej. Na gymech se může spawnout (objevit) raid (typ bitvy) s určitým Raid Bossem (velmi silný pokémon s výrazně nadsazenými statistikami). Nejprve se odhalí vajíčko, ze kterého se v určitý čas vyklube Raid Boss. Na jeho poražení a následné získání je potřeba jednoho až více hráčů (podle obtížnosti), kteří se musí domluvit na konkrétním čase, dojít na místo (případně dostat pozvánku a účastnit se vzdáleně) a společně bosse porazit. Po úspěšné bitvě dostanou hráči různé odměny v podobě stardustu berries, dalších předmětů do hry a možnost chytit poraženého pokémona. Z pokéstopů hráč dostává úkoly, z kterých může dostat další pokébally, berries nebo přímo spawn pokémona. Hráči mohou sílu svých nasbíraných pokémonů změřit v PvP nebo PvE souboji. Pokud se hráči osobně setkají, mohou své pokémony vyměňovat. Hráči jsou rozděleni do tří týmů – modrý Mystic, červený Valor, žlutý Instinct. Hra je vyvíjena společností Niantic.

2.1 Ingress

Ingress je mobilní hra podobná hře Pokémon GO. Spravuje ji stejný vývojář – Niantic. Ve hře jsou 2 týmy neboli frakce (zelení Osvícení a modrý Odpor), kteří bojují proti sobě. Místo pokéstopů a gymů hráči zabírají portály pro svůj tým a pomocí krystalů se je snaží spojit nejdříve do přímek, poté do geometrického útvaru. Útvar nesmí křížit přímka druhého týmu. Podle velikosti útvaru se týmu a hráčům připisují body. Portály ve hře Ingress se následně promítají do hry Pokémon GO (vysvětleno v kapitole 3.2).

2.2 Motivace

Hlavní motivací k vytvoření tohoto bota je pomoc hráčům při domluvě na Raid Bosse. Komunikace probíhá na Discordu v několika kanálech. Jelikož jde o chatové místnosti, je domluva těžší, protože starší zprávy se snadno ztratí a následně může být náročné se dopočítat, kolik lidí se chce účastnit. To hlavně v případě, kdy začne psát několik lidí najednou. Mezi nimi jsou zkušení hráči i nováčci, a právě pro ty je velmi obtížné se v komunikaci orientovat. Často dochází ke zmatkům, k nedorozumění, někdy může situace dokonce vyústit k nepřiměřenému chování. Někteří hráči nemusí vědět, jak se na domluvené místo dostat, proto by měl bot obsahovat databázi bodů se souřadnicemi a umožnit přehledný výpis bodů s odkazem na mapu pro snadné nalezení. Zjednodušení komunikace a odstranění rozdílů mezi zkušenými hráči a nováčky je jedním z hlavních účelů této práce.

2.3 Discord server

Softwarový robot je testován na soukromém serveru autora práce, který je částečnou kopií hlavního serveru *Pogo Raid Liberec*, kam by měl být bot aplikován. Aby mohl vykonávat svou práci, je nutné mu udělit nejvyšší práva, aby měl přístup ke všem kanálům a mohl do nich psát zprávy či blokovat uživatele a odepsat jim role. Díky komplexnosti Discordu je testovací server plně dostačující a výsledky na něm lze prezentovat jako konečné.

2.4 Textové kanály

Nejdůležitější kanály, které obhazuje testovací i hlavní veřejný server jsou:

- **#hlášení-raidů** pro nahlášení Raid Bosse a informování ostatních hráčů. Hlášení by mělo být na jeden řádek, aby se udržela přehlednost a mělo by obsahovat čas, místo bodu (gymu), pokémona (lze se použít emotikon, pokud to server umožňuje) a odkaz na příslušný kanál oblasti, kam bod spadá. Jakékoli jiné zprávy do tohoto kanálu nepatří.
- **#raid-centrum** je jeden z několika tzv. kanálů pro domluvu. Město Liberec, pod které Discord *Pogo Raid Liberec* spadá, je rozděleno do několika oblastí, aby byla komunikace přehlednější. Dalšími příklady jsou **#raid-staré-město**, **#raid-harcov**. Pro vzdálené raidy na které je nutná pozvánka kanál **#raid-remote**. V případě velkého množství zpráv se mohou kanály kdykoli rozdělit na menší oblasti. V těchto kanálech probíhá samotná komunikace a vzájemná diskuse o možné účasti a v případě dostatečného počtu hráčů i o domluvě srazu. Často se zde trenéři ptají, kde se místo

přesně nachází nebo kdo pošle pozvánku hráčům, kteří nemohou na místo osobně dorazit.

- **#domluvené-srazy** slouží pouze pro napsání domluveného času a místa konání raidů. Zpráva by měla být na jeden řádek. Tímto se dá zároveň vědět ostatním hráčům, kteří se např. nezapojili do výše uvedené diskuse. Kromě srazů je zakázáno psát do kanálů cokoli jiného.
- **#bot-commands** pro využití ostatních funkcí bota, jako např. výpis bodů, je viditelný pro všechny členy. **#bot-admin** je viditelný pouze pro správce a uživatele s oprávněním. Slouží pro administrativní příkazy.

2.5 Systém hlášení raidů

Nejprve hráč ve hře Pokémon Go uvidí raid, na který by chtěl sehnat další hráče. Trenér napíše do příslušného kanálu **#hlášení-raidů** název gymu, název pokémona a čas (může reprezentovat začátek raidu nebo počet minut do konce) a jiný kanál, kde bude probíhat domluva mezi ostatními hráči, ve který čas se uskuteční sraz. Hráči tyto informace mohou zapsat v jakémkoli pořadí. Příklad: čas – pokémon – místo – kanál nebo místo – čas – pokémon – kanál. Název pokémona může být nahrazen emotikonem. Čas může reprezentovat začátek raidu nebo obvykle počet minut do konce. Následně probíhá diskuse s ostatními trenéry, zda je dostatečný zájem, a zvolí se čas, který vyhovuje všem, resp. většině. Po úspěšné domluvě se do kanálu **#domluvné-srazy** napíše místo, pokémon (emotikon) a čas srazu.



Obrázek 3 Ukázka hlášení raidů

2.6 Bot a jeho funkce

Softwarový robot by měl primárně pomoci v automatizaci výše uvedených kroků, aby i pro začátečníky bylo jednoduché systém pochopit. Pro současné hráče, kteří mají ruční systém zažitý, by měl být přechod na práci s botem co nejjednodušší. Základní principy by se neměly tolik lišit od současných.

Bot by měl umět rozpoznat zprávu hlášení raidu a zobrazit ji v přehledné podobě, umožnit uživatelům pomocí emotikonů hlasovat o účasti, a přehledně tak zobrazit počet zájemců. Následně opět pomocí emotikonu vytvořit sraz a vypsát přehlednou zprávu.

2.7 Možné rozšíření

Díky vysokému potenciálu bota a mnoha možnostem jeho využití se nabízí několik možností, jak jeho funkce rozšířit. Patří mezi ně např. odměňování uživatelů za hlášení a účast na raidech. Dále také rozšíření o hlášení úkolů na pokéstopech. Bot by rovněž mohl automaticky blokovat uživatele při porušení pravidel a ukládat jejich historii nebo uchovávat v databázi seznam pokémonů, které hráč vlastní, a umožnit automatické vypsání pokémonů chybějících. Podle zpětné vazby a získaných zkušeností lze tyto rozšíření jednoduše přidat. Tato práce se jimi však dále nezaobírá z důvodu omezeného rozsahu ročníkové práce.

3 Technická pravidla

Tato kapitola se věnuje pravidlům, která jsou ve hře nastavená a pomocí nichž bot může pracovat s daty, dopočítávat a kontrolovat data.

3.1 S2 buňky

S2 buňky fungují pomocí matematických výpočtů, které umožňují převádět 3D tvar Země do 2D geometrie. Technologii vynalezl v roce 2005 inženýr Googlu Eric Veatchem. Každá buňka má jedinečné 64bitové číslo, které umožňuje vývojářům jednoduchou identifikaci míst na Zemi. Mají různé úrovně podle velikosti (vyšší úroveň = menší velikost). Jsou hierarchické, větší buňku lze rozdělit na menší. Největší buňka úrovně 0 má rozměr 85 011 012 km², naopak nejmenší buňka úrovně 30 má rozlohu 0,48 cm² až 0,93 cm².¹⁰

Pokémon GO využívá tyto buňky v několika úrovních. Buňky úrovně 10 se používají k určení polohy při chycení pokémona. Úrovně 10 a 11 slouží k určení počasí ve hře. Úroveň 14 se používá k určení počtu gymů v dané buňce. Body, které jsou v úrovni 16 kolem aktuální polohy hráče, se vykreslují ve hře. Buňky úrovně 17 se používají k určení, zda se bod objeví jen ve hře Ingress, nebo i ve hře Pokémon GO.¹¹

3.2 Pravidla zobrazení bodů ve hře

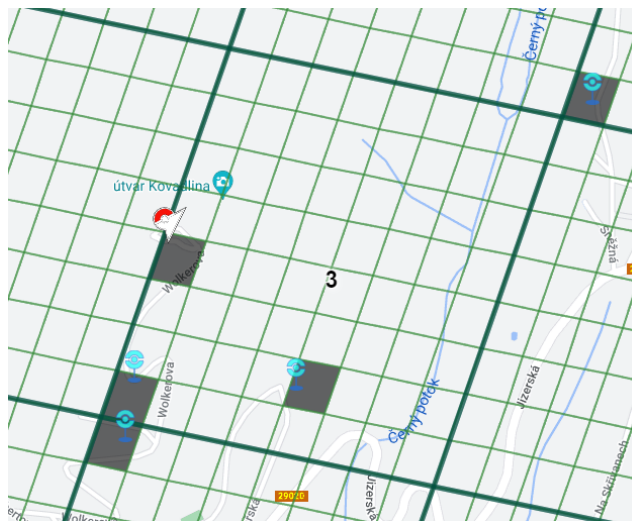
Pokéstopy a gymy mají určitá pravidla, podle kterých se ve hře objevují. Většina bodů (portálů) vychází ze hry Ingress. Dříve se nové body mohly přidávat pouze přes něj. Před dvěma lety se umožnila možnost přidávat body i ze hry Pokémon GO.

V buňce úrovně 17 může být ve hře Pokémon GO pouze jeden pokéstop nebo gym nehledě na počet portalů ve hře Ingress. Na obr. č. 3 jsou znázorněny světle zelenou barvou. V buňce úrovně 14 na obr. č. 3 znázorněny silnou zelenou čarou. Podle počtu zaplnění buněk úrovně 14, kterých je 64, se vypočítává počet gymů. Přesněji řečeno, při zaplnění 2–5 buněk úrovně 17 (na obr. č. 3 znázorněno šedým vyplněním) se jeden z bodů změní na gym.

¹⁰ PROCOPIUC, Octavian. *Geometry on the Sphere* [online]. [cit. 2022-08-22]. Dostupné z: https://docs.google.com/presentation/d/1Hl4KapfAENAO4gv-pSngKwvS_jwNVHRPZTTDzXXn6Q/view

¹¹ ZEROGHAN. A Comprehensive Guide to S2 Cells and Pokémon GO. In: *Pokémon GO Hub* [online], 27. 2. 2019 [cit. 2022-08-22]. Dostupné z: <https://pokemongohub.net/post/article/comprehensive-guide-s2-cells-pokemon-go/>

Při vyplnění 6–19 buněk 2 gymy. Pokud bude počet bodů mezi 20–34, v buňce budou 3 gymy.¹² Na obr. č. 3 je vidět, že v buňce úrovně 14 jsou vyplněná 3 políčka, tudíž se bod v jednom z nich změní na gym. Změní se vždy pokéstop, který má nejvíce lajků ve hře.



Obrázek 4 Mapa znázorňující body ve hře a ohrazení S2 buněk úrovně 14 a 17

3.3 Získání dat o bodech

Ingress ani Pokémon GO nemají žádné oficiální API, ze kterého by se dala získat data o bodech ve hře. Existují webové stránky, kde mohou lidé ručně body přidávat. Nevýhodou těchto stránek je náročné ověřování informací a neúplnost dat. Pro řešení se nabízí oficiální Ingress mapa, která je dostupná pro přihlášené uživatele na adrese www.intel.ingress.com. V ní jsou vidět všechny portály ve hře a jaký tým je má pod kontrolou. Data ve výchozím stavu stále nejdou exportovat. Je nutné použít doplněk do prohlížeče *Ingress Inter Total Conversion* (dále IITC). Pomocí něj lze jednotlivé portály označit za pokéstopy nebo gymy a následně exportovat do souboru .csv nebo .json. Pro aplikaci je využit soubor .csv, který obsahuje 3 nejpodstatnější informace – jméno a zeměpisná délka a šířka.

¹² ZEROGHAN. A Comprehensive Guide to S2 Cells and Pokémon GO. Pokémon GO Hub [online], 27. 2. 2019 [cit. 2022-08-22]. Dostupné z: <https://pokemongohub.net/post/article/comprehensive-guide-s2-cells-pokemon-go/>

| | A | B | C |
|----|-------------------------|-----------|-----------|
| 43 | Puzzle Statue | 50.783651 | 15.061623 |
| 44 | Trafika | 50.784754 | 15.051917 |
| 45 | Umění v Lidových Sadech | 50.777981 | 15.082147 |
| 46 | Popraskaný Obličej | 50.758614 | 15.051462 |
| 47 | Playground Letka | 50.775543 | 15.040782 |
| 48 | Mojžiš | 50.771454 | 15.050019 |
| 49 | Colosseum - socha | 50.780863 | 15.049486 |
| 50 | Dětské hřiště | 50.780641 | 15.047558 |
| 51 | Spící žena | 50.779983 | 15.056911 |
| 52 | Horní kasarna | 50.781499 | 15.070168 |
| 53 | Dětské Hřiště Veřejné | 50.78527 | 15.069879 |

Obrázek 5 Ukázka exportovaného souboru pomocí rozšíření IITC

3.4 Zpracování dat o bodech

Získaná data bot zpracuje a vytvoří z nich databázi bodů, ke kterým přidá nové informace, které lze dopočítat nebo přidat ručně. Ze souřadnic lze vypočítat unikátní číslo podle potřebné úrovně (více v kapitole 3.1), automaticky se vygeneruje datum a čas vytvoření a aktualizace. Pokud vznikne konflikt, přidá se příznak nutné ruční kontroly a poznámka. Ručně se musí vyplnit, zda se jedná o pokéstop, nebo gym.

3.5 Vznik konfliktů

Při vkládání bodů do databáze může dojít ke konfliktu, který není možné vyřešit automaticky, ale je nutný ruční zásah správce. Ve hře Pokémon GO totiž nejsou názvy pokéstopů a gymů unikátní. Určité místo je častokrát jedinečné pouze v blízkém okolí a lze předpokládat, že se může objevit i v jiné části města/oblasti. Unikátnost je zaručena souřadnicemi a již zmíněnými S2 buňkami úrovně 17 (kapitola 3.1). Body před zapsáním do hry prochází procesem kontroly, kde správce kontroluje, zda se nejedná o duplikát. Protože body vytvářejí a schvalují lidé, tak často nemusí být názvy gramaticky správně (např. chybná interpunkce nebo nedodržení pravopisu velkých písmen u názvů). Problémem jsou příliš obecné názvy bez bližší specifikace místa. Nejčastějšími problematickými názvy jsou: *Dětské hřiště*, *Graffiti*, *Památník*, *Pískoviště*, *Venkovní posilovna*.

Bot musí vzniklé konflikty odchytit, odhadnout o jaký konflikt se jedná a zapsat je do databáze pro pozdější ruční kontrolu. Pro případy, kdy se nepovede konflikt odhalit, musí bot umět pomocí příkazu změnit název, souřadnice nebo typ bodu a aktualizovat databázi. Pro kontrolu zapsaných bodů by měl umět vypsát list bodů v databázi po napsání názvu nebo části názvu.

4 Implementace aplikace

4.1 Databáze

V databázi potřebujeme ukládat informace o bodech a pokémonech. Každá entita má primární klíč (Id) a atribut vytvoření a aktualizace záznamu. Dalšími atributy v tabulce *Points* jsou: název bodu, typ (portál, pokéstop, gym), zeměpisná délka a šířka, identifikátor buňky S2 úrovně 14 a 17, položka nutnosti kontroly záznamu a informace o chybě. Entita *Pokemons* obsahuje všechny potřebné informace o pokémonech, viz Obrázek 6. I když pro využití v této práci nám postačuje pouze název a odkaz na obrázek, je dobré mít další informace již uložené pro možné budoucí rozšíření. Databáze SQLite je uložena v souboru *PogoLiberec.db*.

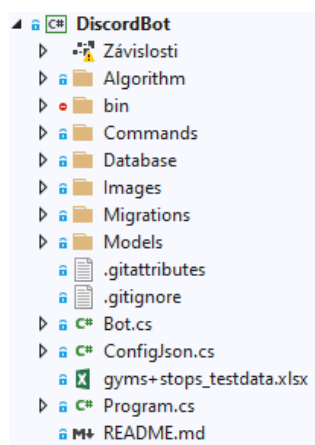
| Points | Pokemons |
|-------------------------|---------------------------|
| PointId INTEGER NN (PK) | PokemonId INTEGER NN (PK) |
| Name TEXT NN | Name TEXT NN |
| Type INTEGER NN | ImageUrl TEXT |
| Latitude REAL NN | PokedexNumber INTEGER NN |
| Longitude REAL NN | Type1 INTEGER NN |
| IdCell14 TEXT NN | Type2 INTEGER NN |
| IdCell17 TEXT NN | Event TEXT |
| NeedCheck INTEGER NN | Shiny INTEGER NN |
| CheckedInfo TEXT | Tradable INTEGER NN |
| CreatedAt TEXT NN | Legendary INTEGER NN |
| UpdatedAt TEXT NN | Generation INTEGER NN |
| | Mythical INTEGER NN |
| | Regional INTEGER NN |
| | Form INTEGER NN |
| | MegaEvolution INTEGER NN |
| | RegionalArea TEXT |
| | Release TEXT |
| | CreatedAt TEXT NN |
| | UpdatedAt TEXT NN |

Obrázek 6 Schéma databáze

4.2 Souborová struktura

Soubory programu jsou rozděleny do několika kategorií. Složka *Algorithm* obsahuje třídy pro práci s daty. Složka *bin* slouží pro zdrojové soubory softwaru v provozu. Ve složce *Commands* se nachází třídy obsluhující příkazy bota. Ve složce *Database* je *PogoContext.cs*, který zajišťuje komunikaci mezi kódem a databází. Ve složce *Images* jsou uloženy obrázky pokémonů pro možné pozdější využití. Bot využívá pro obrázky data z internetu. Složka *Migrations* obsahuje vygenerované migrační soubory. Ve složce *Models* se nachází modely, jako například *Points.cs* nebo *Pokemons.cs*. Soubor *Bot.cs* je třída, která reprezentuje instanci

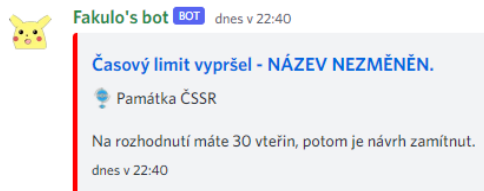
bota. Soubor *ConfigJson.cs* obsahuje předpis konfiguračních dat, konkrétně vlastnosti *Token* a *Prefix*, které slouží pro autentizaci bota. *Program.cs* je hlavní spouštěcí soubor aplikace. *README.md* je vytvořen GithHubem a slouží pro vložení důležitých informací. Soubor *.gitignore* definuje, které soubory nebo složky se nebudou na GitHub nahrávat, protože při nasazení do provozu se vygenerují automaticky. *Gym+stops_data.xlsx* jsou vstupní data, s kterými bot dále pracuje.



Obrázek 7 Souborová struktura programu

4.3 Příkazy bota

Všechny příkazy, na které bot reaguje musí začínat prefixem, který je nastavený na vykřičník (!). Výjimku tvoří hlášení a domlouvání raidů z důvodu vyššího komfortu pro uživatele, aby byl přechod na práci s botem co nejjednodušší. Pro příkazy bez prefixu je nastaven konkrétní kanál, aby bot dokázal komunikaci rozpoznat. Na příkazy, které jakkoli přidávají data do databáze nebo je jakkoli mění či odstraňují, bot odpovídá pouze v předem nastaveném kanálu, ke kterému mají přístup pouze správci. Aby se předešlo problémům s čekáním na odpověď, tak je stanoven limit 30 sekund pro potvrzení změn. Limit může být v budoucnu změněn podle aktuálních potřeb a zpětné vazby uživatelů.



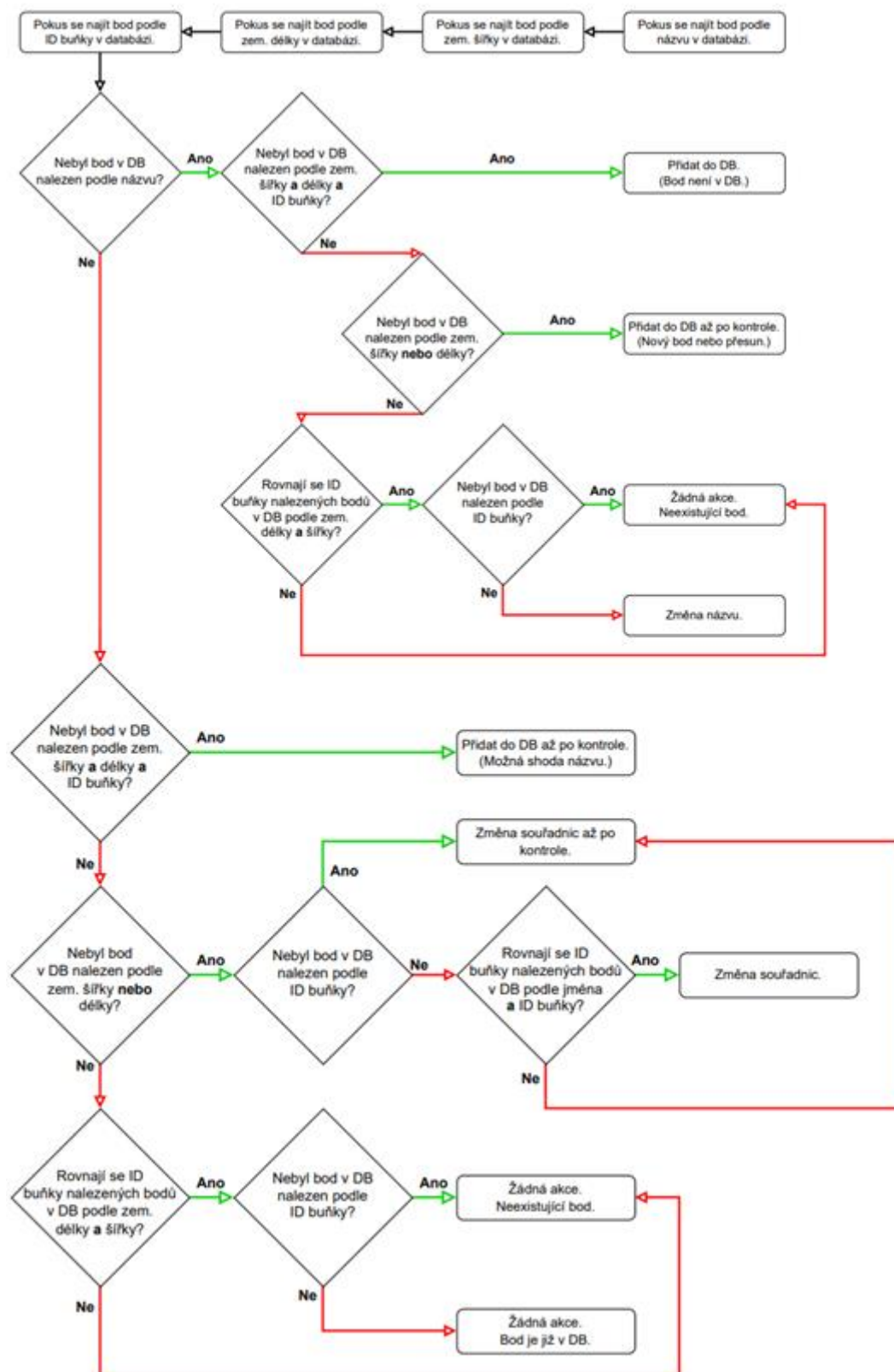
Obrázek 8 Ukázka zprávy při vypršení limitu

4.3.1 Nahrávání dat

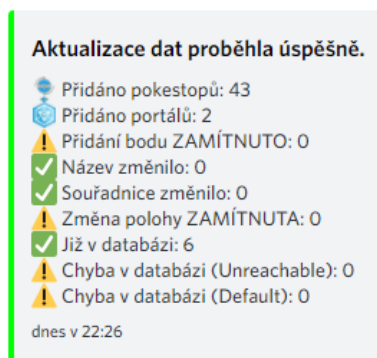
Vstupní data byla získána z mapy hry Ingress pomocí rozšíření IITC viz kapitola 3.3. Soubor ve formátu .csv musí být vložen do kořenového adresáře pod jménem *gym+stops_data*. Následně se rozparsuje po řádcích. Tím se vytvoří nový list bodů připravených pro vložení do databáze. Výsledek se dále postupně zpracovává ve for cyklu. Nejdůležitější částí je kontrola, v jakém stavu je vstupní bod s již vytvořenou databází, viz kapitola 3.5. Vývojový diagram zpracování zobrazuje Obrázek 10. Nejprve se pokusí najít bot podle každé informace, kterou známe (název, zeměpisná šířka a délka a S2 číslo buňky – v diagramu jako ID buňky). Pokud ani jedna informace není v databázi, bod se přidá. V případě, kdy chybí jen jedna či obě souřadnice, vypíše se správci zpráva o možném konfliktu. Ten musí pomocí emotikonu rozhodnout, zda se bod přidá nebo ne. Jestliže je rozdíl pouze název, změní se v databázi na nový. Když se shoduje název a jedna souřadnice, došlo ke změně místa a je opět nutná kontrola správcem. Pokud jsou všechny informace v databázi, jedná se o duplikát, s kterým není nijak dále zacházeno. Metoda vrací výčtový typ *State* (Obrázek 9), který informuje o výsledku zpracování. Nahrání nových bodů se spustí po zadání příkazu **!addData**. Každý zápis do databáze je také vypsán do chatu a na konci nahrávání se vypíše přehledná tabulka s počty o přidaných bodech viz. Obrázek 11.

```
Počet odkazů: 20
public enum State
{
    AddToDB,
    AddToDBCheck,
    ChangeName,
    ChangeNameCheck,
    ChangePosition,
    ChangePositionCheck,
    ChangePositionOrAdd,
    Duplicate,
    Unreachable
}
```

Obrázek 9 Výčtový návratový typ



Obrázek 10 Vývojový diagram postupu vyhodnocení existence bodu



Obrázek 11 Výpis zápisu bodů

4.3.2 Kontrola dat

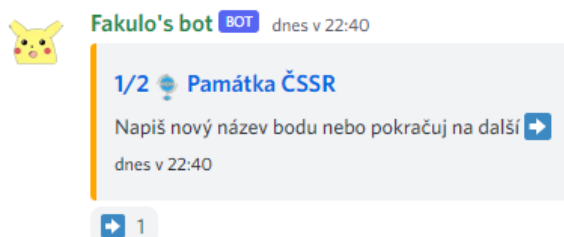
Po napsání příkazu **!check** se vypíší všechny body, které mají nastavenou vlastnost *NeedCheck* na výčtový typ *Yes*. Správce následně bude mít možnost změnit úplně vše (název, typ, souřadnice). Po úspěšné úpravě a zapsání do databáze se změní *NeedCheck* na *Checked* a do poznámky se vloží, co bylo ručně změněno.



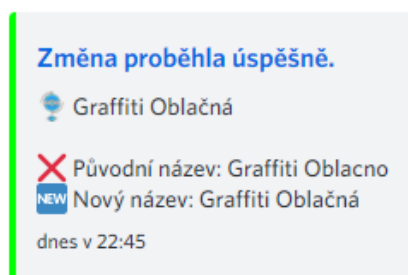
Obrázek 12 Zpráva v případě, že nejsou žádné body na kontrolu.

4.3.3 Změna názvu bodu

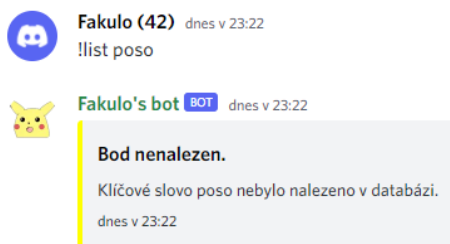
Pomocí příkazu **!changeName mezera název** lze změnit název bodu. V databázi se vyhledají všechny body, které obsahují celé klíčové slovo nebo i jeho část. Správci se následně zobrazí informace o nalezení a možnost zadání nového názvu (Obrázek 13). Pro přehlednost je vidět aktuální stránka a celkový počet. Správce tedy může zadat nový název nebo pomocí emotikonu přeskočit na další. Po časovém limitu se změna automaticky zamítné a pokračuje se na další.



Obrázek 13 Zpráva bota pro změnu názvu



Obrázek 14 Potvrzení změny názvu



Obrázek 15 Zpráva, kdy není nalezen žádný bod

4.3.4 Změna typu bodu

Pomocí příkazu **!changeType mezera název** lze změnit typ bodu. Postup je totožný se změnou jména viz kapitola 4.3.3. Místo napsání nového názvu, má správce na výběr z emotikonů, kterými typ změní. Emotikon gymu, EX gymu (speciální druh gymu, který je v současné době ve hře nevyužíván), pokéstop a portál. Emotikon aktuálního stavu bodu není zobrazen (na obrázku č. Obrázek 16 není zobrazen emotikon pokéstopu). Dále lze pomocí šipky přeskočit na další bod.

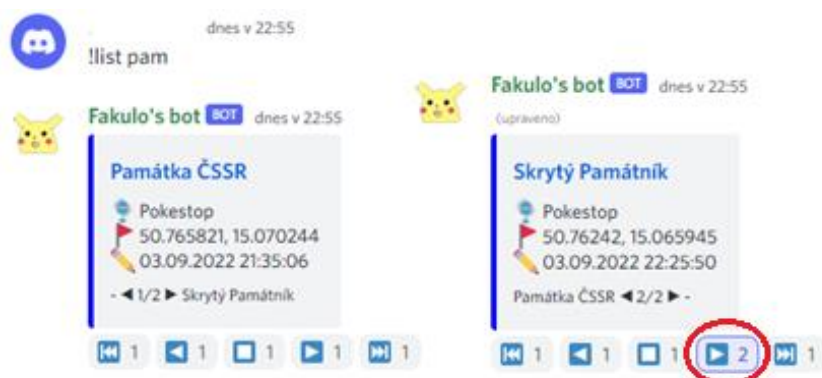


Obrázek 16 Výběr nového typu bodu

4.3.5 Procházení bodů

Přes příkaz **!list mezera název**, se hráči vypíše stránkovaný seznam všech nalezených bodů a klíčovým názvem. V tabulce uživatel vidí název a odkaz na mapu, typ, souřadnice a poslední úpravu. Ve spodní části pak předchozí, následující bod a aktuální a celkový počet

stránek. Přepínání mez stránkami lze provést pomocí emotikonů, které nahrazují šipky. Stisknutím emotikonu čtverce neboli stop se zpráva smaže.



Obrázek 17 Ukázka stránkového zobrazení hledaných bodů

4.3.6 Hlášení raidů

Hlášení raidu funguje pouze v kanále `#hlášení-raidů`, a to bez použití prefixu. Bot každou zprávu zpracuje a pokusí se nalézt všechny potřebné informace. V případě nalezení informace je vzápětí ze zadaného textu odebrána, aby bylo jednodušší analyzovat zbylý text a s větší pravděpodobností rozpoznat všechny informace. Nejprve je v textu vyhledán kanál pro diskuzi. Následuje hledání emotikonu, který může být použit místo názvu pokémona. Jako další se hledá čas líhnutí nebo zbývajíc čas, poté se bot snaží najít pokémona. Zbylý text, se kterým se pracuje, se rozdělí na jednotlivá slova a ta se hledají v databázi *Pokemons*. V případě pouze jednoho výsledku pokračuje proces dál na hledání místa (gymu). Text, který zůstal se rozdělí pomocí mezery na jednotlivá slova, která se potom hledají v databázi bodů. Četnost výsledků se seřadí podle nejvyšší, a tím se získá nejpravděpodobnější název gymu. Jestliže bude četnost stejná pro více bodů nebo jakákoli jiná informace nenalezena, bude tato skutečnost vypsána hráči, aby se pokusil napsat hlášení raidů znovu a aby si dal pozor na uvedené informace.



Obrázek 18 Zpráva nahlášení raidu

4.3.7 Ukázání zájmu

Poté co uživatel správně nahlásí raid, bot zprávu úspěšně zanalyzuje a vypíše do příslušného kanálu pro diskuzi zprávu o informacích raidu, které získal. Konkrétně čas, název a obrázek pokémona, název gymu a celkový počet zájemců a seznam jejich přezdívek. Jedná se o aktivní odkaz, aby bylo případné kontaktování co nejjednodušší. Pro ostatní uživatele je jednodušší postřehnout nově nahlášený raid. Mají možnost reakce pomocí emotikonu, zda se chtějí účastnit osobně, potřebují pozvat anebo vědí o dalších účtech, které nejsou součástí Discord komunity (emotikon čísla). Svoji reakci můžou jakkoli měnit nebo zrušit emotikonem křížku. Samotná zpráva se díky tomu automaticky mění. Obrázek 19 znázorňuje příklad, kdy 2 hráči projevili zájem a jeden z nich ví o 2 dalších trenérech, kteří se chtějí účastnit, ale nejsou na Discordu.



Obrázek 19 Přehled raidu a hráčů, kteří mají zájem

4.3.8 Nastavení srazu

Až se hráči domluví na konkrétním čase, může jeden z nich nahlásit sraz pomocí emotikonu palce nahoru ve zprávě, kterou napsal bot. Po stisknutí emotikonu bot vyzve k napsání času srazu viz Obrázek 20. Robot hledá pouze čas ve formátu hh:mm. Ostatní formáty jsou ignorovány. Po napsání času se automaticky do kanálu #domluvené-srazy napíše zpráva o oficiálním srazu, pokémon a místo raidu s odkazem na mapu pro jednodušší nalezení, viz Obrázek 21.



Obrázek 20 Nastavení času srazu



Obrázek 21 Zpráva domluveného srazu

Závěr

Softwarový robot, o kterém tato práce pojednává, splňuje požadavky, které byly v zadání požadovány. Bot dokáže zpracovat vstupní data a vytvořit databázi bodů v Liberci, kterou následně udržuje. Robot umožňuje uživatelům Discordu přehledné zobrazení bodů z databáze s možností otevření bodu v mapách společnosti Google. Správci serveru mohou tuto databázi upravovat přímo pomocí chatu. Konkrétně mohou měnit název, typ a souřadnice bodů. Zároveň vidí i vzniklé konflikty, které mohou nastat při zpracování dat. Bot urychluje a usnadňuje komunikaci ohledně hlášení raidů a domluvě času srazu. Uživatelé vidí vše přehledně a bez nutnosti téměř vše ručně vypisovat.

Jelikož je bot stále ve vývoji, je nasazen na testovacím serveru řešitele této práce. Server je věrnou kopií oficiálního serveru Pogo Raid Liberec, kde bude později po důkladném testování a proškolení stávajících uživatelů přidán. Bot je připraven na několik rozšíření, které v budoucnu pomohou komfort uživatelů ještě zvýšit.

Literatura

ALBAHARI, Joseph a Eric JOHANNSEN. *C# 8.0 in a Nutshell*. 1. vyd. Sebastopol: O'Reilly, 2020. ISBN 978-1492051138.

DB Browser for SQLite [online]. [cit. 2022-08-20]. Dostupné z: <https://sqlitebrowser.org/about/>

Discord [online]. [cit. 2022-08-20]. Dostupné z: <https://discord.com/safety/360044149331-What-is-Discord>

DSharpPlus Contributors [online]. [cit. 2022-08-21]. Dostupné z: <https://dsharpplus.github.io/>

HERCEG, Tomáš. Úvod do .net frameworku. In: *Dotnetportal.cz* [online]. 3. 4. 2009 [cit. 2022-08-20]. Dostupné z: <https://www.dotnetportal.cz/clanek/125/Uvod-do-NET-Frameworku>

Microsoft [online]. [cit. 2022-08-20]. Dostupné z: <https://www.microsoft.com/cs-cz/>

PROCOPIUC, Octavian. *Geometry on the Sphere* [online]. [cit. 2022-08-22]. Dostupné z: https://docs.google.com/presentation/d/1HI4KapfAENAO4gv-pSngKwvS_jwNVHRPZTTDzXXn6Q/view

SQLite [online]. [cit. 2022-08-20]. Dostupné z: <https://www.sqlite.org/about.html>

ZEROGHAN. A Comprehensive Guide to S2 Cells and Pokémon GO. In: *Pokémon GO Hub* [online], 27. 2. 2019 [cit. 2022-08-22]. Dostupné z: <https://pokemongohub.net/post/article/comprehensive-guide-s2-cells-pokemon-go/>

Seznam obrázků

| | |
|--|----|
| Obrázek 1 Grafické rozhraní testovacího serveru na Discordu..... | 11 |
| Obrázek 2 Grafické rozhraní programu DB Browser | 12 |
| Obrázek 3 Ukázka hlášení raidů..... | 15 |
| Obrázek 4 Mapa znázorňující body ve hře a ohraničení S2 buněk úrovně 14 a 17 | 18 |
| Obrázek 5 Ukázka exportovaného souboru pomocí rozšíření IITC | 19 |
| Obrázek 6 Schéma databáze..... | 20 |
| Obrázek 7 Souborová struktura programu | 21 |
| Obrázek 8 Ukázka zprávy při vypršení limitu..... | 21 |
| Obrázek 9 Výčtový návratový typ | 22 |
| Obrázek 10 Vývojový diagram postupu vyhodnocení existence bodu | 23 |
| Obrázek 11 Výpis zápisu bodů..... | 24 |
| Obrázek 12 Zpráva v případě, že nejsou žádné body na kontrolu. | 24 |
| Obrázek 13 Zpráva bota pro změnu názvu..... | 25 |
| Obrázek 14 Potvrzení změny názvu..... | 25 |
| Obrázek 15 Zpráva, kdy není nalezen žádný bod | 25 |
| Obrázek 16 Výběr nového typu bodu | 25 |
| Obrázek 17 Ukázka stránkového zobrazení hledaných bodů..... | 26 |
| Obrázek 18 Zpráva nahlášení raidu..... | 26 |
| Obrázek 19 Přehled raidu a hráčů, kteří mají zájem | 27 |
| Obrázek 20 Nastavení času srazu..... | 28 |
| Obrázek 21 Zpráva domluveného srazu..... | 28 |