



## **Středoškolská technika 2022**

**Setkání a prezentace prací středoškolských studentů na ČVUT**

### **PET BOT**

**Tobiáš Horváth**

**Střední škola informatiky a finančních služeb, Plzeň,  
Klatovská tř. 2778/200G, 301 00 Plzeň**

### **Anotace**

Tato práce SOČ se zabývá vytvořením funkční hry pomocí Discord bota. Obsahuje teoretickou část, která popisuje komunikační platformu Discord, knihovny pro programování botů, potřebné systémy pro vytvoření Discord bota a tvorbu Discord bota dle vlastní konfigurace. Aplikační část obsahuje základní příkazy pro bota a popisuje bota z technického hlediska, včetně využívání API, protokolu WebSocket a databáze pro ukládání dat o uživatelích a mazlících.

### **Klíčová slova**

Discord bot; interaktivní hra; knihovny pro programování; API; WebSocket.

## Obsah

Úvod .....	3
1. Teoretická část.....	4
1.1 Možné komunikační platformy podporující boty.....	4
1.1.1 Discord .....	4
1.1.2 Telegram.....	5
1.1.3 Facebook .....	5
1.1.4 Twitter .....	6
1.2 Knihovny platformy Discord.....	7
1.2.6 Po dlouhém rozhodování – Discord.js.....	8
1.3 Potřebné věci pro fungování discord bota a jak vytvořit dle vlastní konfigurace .....	9
1.3.1 Potřebné věci pro funkčnost bota .....	9
1.3.2 Vytvoření bota dle vlastní konfigurace .....	9
2. Aplikační část.....	11
2.1 Základní příkazy a informace pro využívání bota.....	11
2.2 PetBot z IT hlediska .....	16
2.2.3 Databáze a ukládání dat o uživateliích .....	17
2.2.4 Struktura příkazů bota .....	18
Závěr.....	21
Použitá literatura.....	23
Seznam Obrázků.....	24
1.	

# Úvod

V rámci mé odborné práce jsem se rozhodl vytvořit dětskou hru za použití Discord Bota pro platformu Discord pomocí knihovny Discord.js. Záměrem bylo vytvořit zábavnou a interaktivní hru pro uživatele, kterou by mohli hrát i bez vyšších nároků na parametry zařízení.

Discord je populární komunikační platforma používaná zejména mezi hráči počítačových her, ale s rostoucí popularitou se stále častěji používá i jako nástroj pro komunikaci mezi přáteli nebo pro vzdělávací účely. Využití Discordu na vytvoření hry pro uživatele se mi jevilo jako zajímavé a inovativní řešení. Knihovna Discord.js pro mě byla jasnou volbou, neboť se jedná o nejoblíbenější knihovnu pro vývoj botů pro Discord a nabízí širokou škálu možností tvorby interaktivních aplikací.

Během procesu vývoje hry jsem se snažil zohlednit požadavky a očekávání cílové skupiny, tedy věkové skupiny v rozmezí od 16 do 20 let. To znamenalo zaměřit se na zábavný a přístupný herní koncept, ale zároveň také zajistit, aby hra obsahovala „grind“ prvky. Výsledkem mé práce je hra nazvaná "Pet Bot", ve které se hráči starají o své virtuální mazlíčky. Cílem hry je chovat několik druhů mazlíčků, krmit je, nakupovat jim pamlsky, hrát si s nimi nebo je také vyměňovat, přičemž hráči postupně své mazlíčky vylepšují.

Věřím, že hra "Pet Bot" představuje zajímavou formu online hry, která uživatelům Discordu zároveň umožní si částečně vyzkoušet, co by obnášelo se starat o reálné mazlíčky. Jsem přesvědčen, že vytvoření hry pro platformu Discord nabízí skvělý přístup k zábavě pro uživatele, a doufám, že hra bude úspěšná.

# 1. Teoretická část

## 1.1 Možné komunikační platformy podporující boty

Kapitola pojednává o možných platformách, mezi kterými jsem se rozhodoval při výběru prostředí pro vytvoření bota. Uvedu zde výhody a nevýhody jednotlivých platform a důvod výběru platformy Discord.

### 1.1.1 Discord

Výhody programování bota pro platformu Discord:

**Přístupnost a dostupnost:** Bot je k dispozici 24/7 a kdekoliv, kde je k dispozici internetové připojení, což umožňuje hráčům přístup ke hře nebo komunikaci s botem, kdykoli je jim to pohodlné.

**Široké možnosti personalizace:** Bot může být přizpůsoben konkrétním potřebám a požadavkům uživatele a může být integrován s různými aplikacemi a službami.

**Možnost vytváření interaktivních aplikací:** Boti pro Discord jsou schopni reagovat na příkazy a zprávy uživatelů a posílat zpět odpovědi nebo provádět určité akce. To umožňuje vytvářet velmi interaktivní aplikace, jako jsou hry, kvízy, soutěže atd.

**Možnost automatizace:** Boti pro Discord mohou být nastaveni tak, aby automaticky reagovali na určité příkazy nebo události, což může ušetřit čas a námahu při řešení rutinních úkolů.

Nevýhody programování bota pro platformu Discord:

**Náročnost na vývoj:** Vytvoření bota pro Discord může být pro některé vývojáře náročné, zejména pokud nemají dostatečné znalosti o programování nebo o knihovně Discord.js.

**Omezené možnosti:** Ačkoliv knihovna Discord.js nabízí širokou škálu možností, stále existují určitá omezení, která se týkají možností tvorby možností tvorby aplikací za pomoci Discord botů.

**Bezpečnost:** Je důležité dbát na bezpečnost při vývoji bota a zajistit, aby nebyl zneužit k šíření škodlivého kódu nebo k jiným nelegálním činnostem.

**Náročná údržba a aktualizace:** Vyžaduje pravidelné aktualizace a opravy, aby byl bot funkční a bez chyb. Discord pravidelně aktualizuje své funkce, a proto knihovny často upravuje své syntaxe.

## 1.1.2 Telegram

Výhody programování bota pro platformu Telegram:

**Snadné použití:** Programování bota pro Telegram je snadné díky přístupnému API a dobře dokumentovaným nástrojům pro vývojáře.

**Různorodost funkcí:** Boti pro Telegram jsou schopni provádět různorodé úkoly od posílání zpráv až po automatizaci rutinních úkolů či pracování s online platebními metodami.

**Bezpečnost:** Telegram je známý svým důrazem na bezpečnost a soukromí, což je pro některé uživatele důležité při volbě komunikační platformy.

Nevýhody programování bota pro platformu Telegram:

**Omezené možnosti:** Knihovna pro vývoj botů pro Telegram nabízí menší množství funkcí než jiné knihovny pro vývoj botů, jako je například Discord.js.

**Méně populární platforma:** Telegram není tak populární komunikační platforma jako je například Discord, což může omezit počet potenciálních uživatelů bota.

**Méně vývojářských komunit:** Telegram má menší vývojářské komunity než jiné platformy, což může ztěžovat hledání pomoci nebo řešení problémů během vývoje bota.

## 1.1.3 Facebook

Výhody programování bota pro platformu Facebook:

**Široká základna uživatelů:** Facebook je jednou z nejpobulárnějších sociálních sítí na světě s více než 2 miliardami aktivních uživatelů, v důsledku čehož se na Facebooku nachází velký počet potenciálních uživatelů bota.

**Možnost integrace s dalšími službami:** Boti pro Facebook mohou být snadno integrováni s dalšími službami, například PayPal nebo Slack, díky čemuž může být rozšířena funkčnost botů.

**Vysoká spolehlivost:** Facebook je považován za velmi spolehlivý a je schopen zajistit stabilní fungování botů na své platformě.

Nevýhody programování bota pro platformu Facebook:

**Omezení týkající se soukromí:** Facebook má povinnost dohlížet na soukromí svých uživatelů a může vyžadovat, aby vývojáři splňovali určité podmínky ohledně zacházení s osobními údaji uživatelů.

**Přísné pravidla pro vývojáře:** Facebook má přísná pravidla pro vývojáře, která mohou být pro některé vývojáře obtížná na dodržování.

**Vysoká konkurence:** Na platformě Facebook v současné době funguje velké množství botů, významná konkurence tak může snížit potenciál úspěchu bota.

### 1.1.4 Twitter

Výhody programování bota pro platformu Twitter:

**Rychlá a snadná komunikace:** Twitter je platforma pro rychlou komunikaci. Twitter Boti jsou schopni posílat a přijímat zprávy v reálném čase. Tato funkce může být užitečná pro automatizaci rutinních úkolů kterými jsou například posílání zpráv nebo získávání aktuálních informací.

**Široká základna uživatelů:** Twitter má více než 330 milionů aktivních uživatelů, tento počet značí, že na Twitteru se nachází velké množství potenciálních uživatelů bota.

**Možnost integrace s dalšími službami:** Boti pro Twitter mohou být snadno integrováni s dalšími službami, jako je například Google Maps nebo GitHub, díky nimž je možné rozšiřovat jejich funkčnost.

Nevýhody programování bota pro platformu Twitter:

**Omezený počet znaků:** Twitter má omezený počet znaků, které lze použít ve zprávách, tedy i možnosti funkcí botů naprogramovaných pro Twitter jsou omezené.

**Vysoká konkurence:** Podobně jako na Facebooku, na platformě Twitter je velké množství botů zvyšujících vzájemnou konkurenci, a proto mohou být omezené šance na úspěch nově vytvořeného bota.

**Omezená podpora:** Knihovny pro vývoj botů pro Twitter jsou často hůře dokumentované než knihovny pro jiné platformy, což může ztěžovat vývoj bota.

### 1.1.5 Proč tedy Discord?

Nelze jednoznačně říci, že by Discord byl nejlepší platformou pro vývoj botů. Každá platforma má své vlastní výhody a nevýhody a to, co je pro jednoho vývojáře nejlepší, může být pro jiného nevhodné. Discord se může zdát jako atraktivní volba pro vývoj bota zejména díky své oblíbenosti mezi hráči počítačových her a širokému spektru možností, které knihovna Discord.js nabízí. Pro tyto důvody jsem si právě platformu Discord vybral.

## 1.2 Knihovny platformy Discord

Existuje několik knihoven, které lze použít k vývoji botů pro platformu Discord. Mezi nejpoblárnější knihovny patří:

### 1.2.1 Discord.js:

Knihovna pro vývoj Discord botů naprogramovaná v jazyce JavaScript. Je jednou z nejpoblárnějších knihoven pro vývoj botů na Discordu a nabízí širokou škálu funkcí včetně možnosti reagovat na události, odesílat zprávy a přijímat příkazy od uživatelů. Knihovna je snadno použitelná pro vývojáře s různou úrovní zkušeností a je dobře dokumentována.

### 1.2.2 Discord.py:

Knihovna pro vývoj Discord botů naprogramovaná v jazyce Python. Je méně populární než knihovna Discord.py, stále ale nabízí široké spektrum funkcí pro vývoj botů různé úrovně složitosti. Knihovna je snadno použitelná a také dobře dokumentována, což ji činí atraktivní volbou pro vývojáře s různou úrovní zkušeností.

### 1.2.3 Eris:

Knihovna pro vývoj botů pro Discord naprogramovaná v jazyce JavaScript. Je určena pro vývojáře s vyšší úrovní zkušeností a nabízí širokou škálu funkcí pro vývoj složitějších botů. Knihovna je dobře dokumentována a umožňuje vývojářům snadno reagovat na události, odesílat zprávy a přijímat příkazy od uživatelů. Eris také umožňuje vývojářům přizpůsobit vzhled a chování bota a integrovat ho s dalšími službami.

### 1.2.4 JDA (Java Discord API):

Knihovna pro vývoj Discord botů naprogramovaná v jazyce Java. Je snadno použitelná a nabízí široké spektrum funkcí pro vývoj botů různé úrovně složitosti. Knihovna je dobře dokumentována a umožňuje vývojářům snadno reagovat na události, odesílat zprávy a přijímat příkazy od uživatelů. JDA také nabízí možnost přizpůsobit vzhled a chování bota a integrovat ho s dalšími službami.

### 1.2.5 Discordrb:

Knihovna pro vývoj botů pro Discord naprogramovaná v jazyce Ruby. Je snadno použitelná a nabízí široké spektrum funkcí pro vývoj botů různé úrovně složitosti. Knihovna je dobře dokumentována a umožňuje vývojářům snadno reagovat na události, odesílat zprávy a přijímat příkazy od uživatelů, také umožňuje vývojářům přizpůsobit vzhled a chování bota a integrovat ho s dalšími službami.

## 1.2.6 Po dlouhém rozhodování – Discord.js

Rozhodnout, která knihovna bude na programování bota nejvhodnější, nebylo jednoduché. Váhal jsem mezi knihovnou Eris, Discord.js a Discord.py. Discord.py jsem zvažoval, protože jsem Discord bota v jazyce Python ještě neprogramoval a mohla by to být zajímavá zkušenost, přesto jsem tuto myšlenku po čase zavrhnul. Ze dvou knihoven – Eris a Discord.js – jsem se rozhodl právě pro knihovnu Discord.js. Jak jsem uvedl výše, největší výhoda knihovny Eris je její komplikovanost a komplexnost, a proto se za použití této knihovny dají vytvářet složitější boti, avšak knihovna Eris se užívá mnohem méně než Discord.js. Tímto narážím i na největší výhodu knihovny Discord.js, kterou je komunita. Díky větší komunitě Discord.js je knihovna mnohem podrobnější, při jakémkoliv problému se programátor může obrátit na fóra či oficiální Discord server knihovny pro pomoc. To po delším rozhodování byl můj nejdůležitější důvod pro výběr knihovny Discord.js.



## 1.3 Potřebné věci pro fungování discord bota a jak vytvořit dle vlastní konfigurace

Kapitola se zaměřuje na funkční zprovoznění bota na platformě Discord a na jeho případné vytvoření dle vlastní konfigurace.

### 1.3.1 Potřebné věci pro funkčnost bota

Pro to, abychom mohli bota spustit, potřebujeme několik věcí, které se ale s aktualizací Podmínek pro developery od Discordu často mění a přibývají, ale řekneme si zde ty nejdůležitější a neměnné.

*Discord účet:* Discord účet je nezbytný pro vytvoření aplikace a bota na stránkách Discord Developer Portal. Aplikace a účet jsou potřebné pro autentizaci a ověření vašeho bota na Discordu.

*Token:* Token je jedinečný identifikátor vašeho bota, který je třeba použít pro autentizaci a ověření bota na Discordu.

*Knihovna Discord.js:* Knihovna Discord.js je použita pro vývoj tohoto bota, proto je zde uvedena jako podmínka pro funkčnost bota, ale to se mění v závislosti na developerem zvolené knihovně. Discord.js je dostupná prostřednictvím Node.js. Knihovna poskytuje funkce pro komunikaci s Discord API.

*Node.js:* Node.js je platforma pro vývoj aplikací naprogramovaných v jazyce JavaScript, která je potřebná pro použití knihovny Discord.js.

### 1.3.2 Vytvoření bota dle vlastní konfigurace

Tato kapitola se zaměří na funkční zprovoznění bota pomocí knihovny Discord.js dle vlastní konfigurace.

1. Pro účely vytvoření Discord bota je nezbytné zaregistrovat se na Discord Developer Portal a vytvořit si aplikaci a bota, přičemž lze získat token bota.
2. Dále je třeba nainstalovat Node.js na počítač nebo server, kde bude bot hostován, a vytvořit nový projekt Node.js s nainstalovanou knihovnou Discord.js. pomocí příkazu "npm install discord.js" ve vašem command line interface.
3. Poté je třeba vytvořit nový JavaScript soubor a zavolat knihovnu Discord.js pomocí "const Discord = require('discord.js')".
4. Nutné je i vytvoření nového klienta bota pomocí "const client = new Discord.Client()".

Klient bota je třída, která obsahuje všechny funkce a metody pro komunikaci s Discord API a umožňuje vám ovládat bota na Discordu.

5. Nastavte token bota pomocí "client.login(TOKEN)", kde TOKEN slouží jako indikátor toho, že se jedná o vašeho bota.
6. Nyní lze použít různé funkce a metody knihovny Discord.js pro reakci na události, odesílání zpráv a přijímání příkazů od uživatelů. Například lze použít funkci "client.on('message', message => {})" pro reakci na zprávy od uživatelů nebo funkci "message.channel.send('Hello!)" pro odeslání zprávy na Discord kanál.
7. Když jste s vývojem bota hotovi, lze ho spustit pomocí "client.login(TOKEN)" a po dokončení kódu pro bota je nutné jej spustit pomocí příkazu "node bot.js", kde bot.js je název souboru s kódem pro bota. Poté by se měl objevit na Discordu a začít fungovat podle nastavených pravidel a funkcí.
8. Aby byl bot vidět na serveru, je nutné vytvořit speciální odkaz pro pozvání bota na server. Každý bot potřebuje určité povolení, aby mohl např. odesílat zprávy na serveru, stejně jako uživatelé. Proto také u pozvánky musíte zadat veškeré permise, které bot potřebuje.

Vývoj Discord bota může být zábavný a užitečný projekt, který umožní vyzkoušet si práci s API a vytváření interaktivních aplikací pro Discord. Je důležité si ale uvědomit, že Discord boti mohou být používáni za různými účely a mohou mít různé funkce a schopnosti. Můžete vytvořit bota, který bude sloužit jako pomocník pro komunikaci, bota, který bude provádět administrativní úkoly, nebo bota, který bude hrát hry či poskytovat zábavné funkce jako „Pet Bot“. Je také důležité dbát na bezpečnost a ochranu osobních údajů při vývoji bota. Používejte bezpečné metody autentizace a zabezpečení a je třeba dbát na to, abyste nezveřejňovali žádné citlivé informace. Pokud chcete poskytnout vašeho bota veřejně, musíte klást důraz na zákaznickou podporu a aktualizace bota, abyste mohli reagovat na dotazy a připomínky uživatelů a zlepšovat chování a funkce bota.

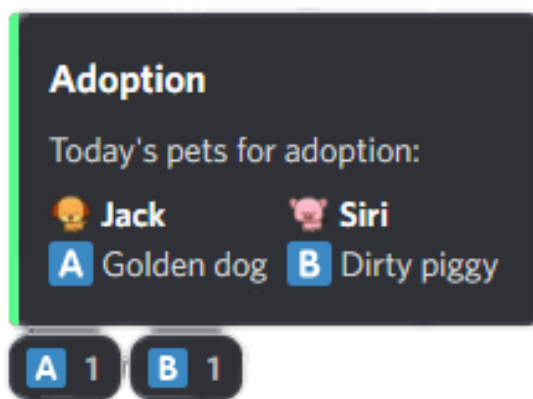
## 2. Aplikační část

Aplikační část mé práce SOČ je hra na principu starání se o mazlíčky fungující přes Discord bota naprogramovaném v jazyce JavaScript pomocí knihovny Discord.js. Bot je založený na adoptování a starání se o svého digitálního mazlíčka pomocí chat příkazů.

Na úvod budou popsány průběhy prvních momentů, když se uživatel poprvé setká s Pet Botem. Poté bude ukázány jeho nejzákladnější funkce a stavba PetBota z IT hlediska; jaké prostředky byly využité z databáze, využité protokoly a stavba příkazů.

### 2.1 Základní příkazy a informace pro využívání bota

Mezi jedním z nejčastějších příkazů u nových uživatelů bývá příkaz „/help“, který vám zobrazí seznam všech příkazů, které mohou uživatelé používat bez speciálních povolení. Bot mezi prvními příkazy zmíní nejdůležitější příkaz ze začátku pro uživatele, a sice „/adopt“. Ten uživateli pošle seznam dvou mazlíčků, kteří mají svá jména a barvu (ke všem druhům zvířat, barevným variacím a dalším informacím o mazlíčcích se dostaneme dále). Tito mazlíčci se mění každých 24 hodin vždy o půlnoci našeho času.



Obrázek 1 Příkaz Adopt

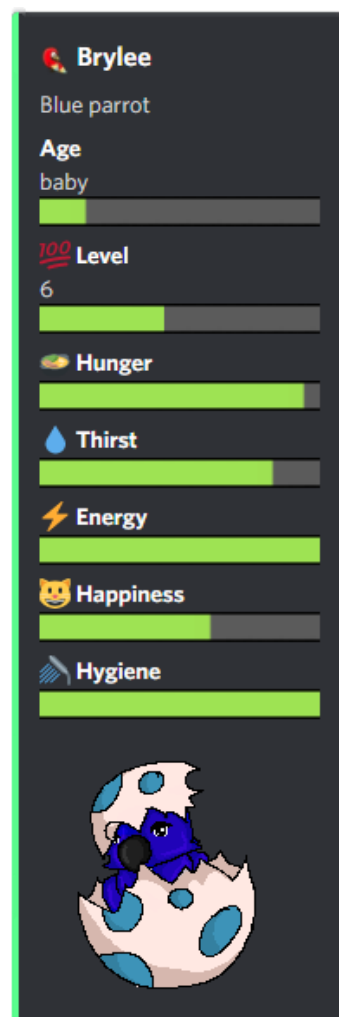
Pomocí reakcí pod zprávou bota si uživatelé mohou vybrat, kterého z mazlíčků si chtějí vybrat a u kterého chtějí zobrazit jeho podrobnosti. Tímto se jim zobrazí podobné okénko jako u příkazy „/mypet“, ve kterém si mohou zobrazit podrobnosti o jimi vlastněném mazlíčkovi. Pokud uživatel má více mazlíčků (limit - 3), dá bot uživateli vybrat podobně jako u příkazu „/adopt“, ale s mazlíčky, které vlastní.

Zde na obrázku je ukázka z příkazu „/mypet“, kde si uživatel vybral svého mazlíčka, Modrého papouška jménem Brylee, který je ve věku mládeže. Uživatel se díky tomuto příkazu může

dozvědět veškeré informace o svém mazlíčkovi, ať se už jedná o jeho věk, level, který se navyšuje s každým příkazem starajícím se o mazlíčka, nebo jeho potřeby – jídlo, pití, energie, radost a hygiena.

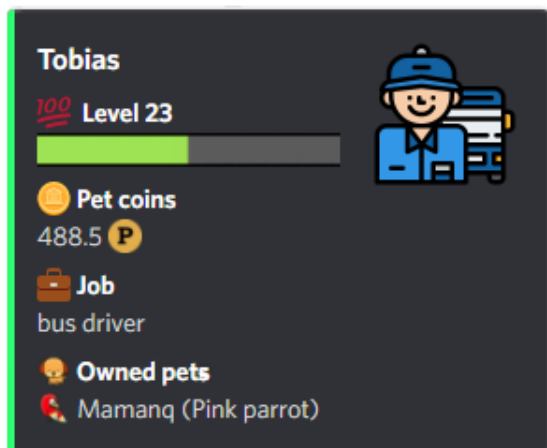
Uživatelé mohou mazlíčky mezi sebou dokonce měnit pomocí příkazu „/trade“. Tento příkaz přináší do bota skvělou mechaniku, díky které mohou lidé mezi sebou měnit např. vzácné mazlíčky nebo mazlíčky s vysokým levelem. U tohoto příkazu byl samozřejmě kladený velký důraz na bezpečnost, aby se zamezilo veškerým problémům týkajících se tzv. scamů.

Další důležitý příkaz je příkaz „/myself“, přes který



Obrázek 2 Příkaz Mypet

si uživatelé zobrazí informace o jejich účtu. Ať už odvedenou práci, Pet Coiny (PetBot měna), jejich level, který jim odemká nové práce vydělávající více Pet Coinů, či odemykají nového

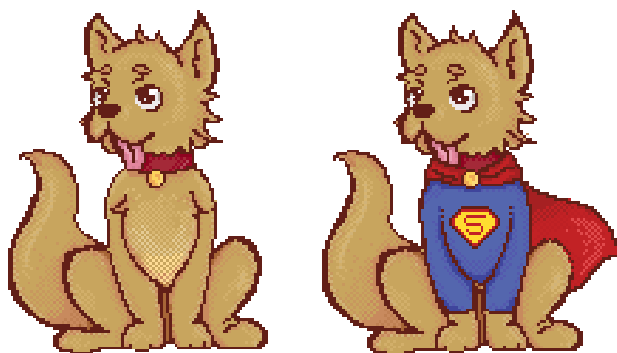


Obrázek 3 Příkaz Myself

vybavení. Tím mohou být pelíšky, díky kterým mazlíčci získávají energii rychleji než při normální spánku, či pamlsky, které neprodleně doplňují hlad, radost a žízeň bez toho, aby přišli o energii jako u příkazů „/feed“, „/drink“ a „/play“.

Na obrázku z odpovědi boti je možné vyčíst uživatelův level (23), počet jeho Pet coinů (488,5), aktuální práci (řidič autobusu) a aktuálně vlastněného mazlíčka (Růžový papoušek se jménem Mamanq).

Pet Coiny hráč může využít v obchodě, který si může otevřít pomocí příkazu „/shop“, kde má několik kategorií, od pamlsků, elixírů, truhel až po pelíšky. Hráč si může kupovat elixíry pro speciální efekty, které mohou způsobovat různé efekty či události, např. instantní dospělost/omládnutí nebo zastavení stárnutí. Truhly jsou zde od získávání vzácných mazlíčků. Vzácní mazlíčci jsou speciální mazlíčci, kteří vypadají jako normální mazlíčci, ale se speciálními kostýmy. Každý zvíře má od začátku jeden speciální skin. V truhlách není speciální skin samozřejmost, vše se točí kolem procentuální šance na získání.



Obrázek 5 Dospělý pes

Obrázek 4 Speciální dospělý pes

Na této stránce jsou ukázány v porovnání dvě možnosti zlatého Retrívra – Normální (nalevo) a Speciální (napravo), který lze získat z truhly, jež si mohou uživatelé zakoupit za Pet Coiny. Uživatelé také mohou jednou za 24 hodin použít příkaz „/daily“ pro každodenní odměnu. Pet Coiny mohou uživatelé vydělávat prací.

Každý 5. level hráče až do levelu 30 odemká 3 nové práce. PetBot

je naprogramovaný tak, tak aby ti lidé, kteří chtějí být aktivnější, si mohli vybrat práci s kratším časem načítání, menším jednorázovým výdělkem, ale celkově vydělají více než v práci, která má sice delší dobu načítání, ale umožňuje získat větší jednorázový výdělek. Cílem je odměnit

aktivnější uživatele, ale zároveň ne poškodit hráče, kteří nemohou být tolik aktivní. Za každou směnu v práci hráči získávají zkušenostní body do levelu, aby se mohli ve hře posouvat.

Samotní mazlíčci od sebe nejsou příliš odlišní. Nejvýznamnější rozdíl je například u rybiček, jež nepotřebují dávat napít příkazem „/drink“, ovšem o to víc Pet Coinů stojí je nakrmit.

Důležité je, že každé zvíře má své vlastní ilustrované modely velikostí, barev a skinů (viz. další strana).



Obrázek 6 Mazlíčci

## 2.2 PetBot z IT hlediska

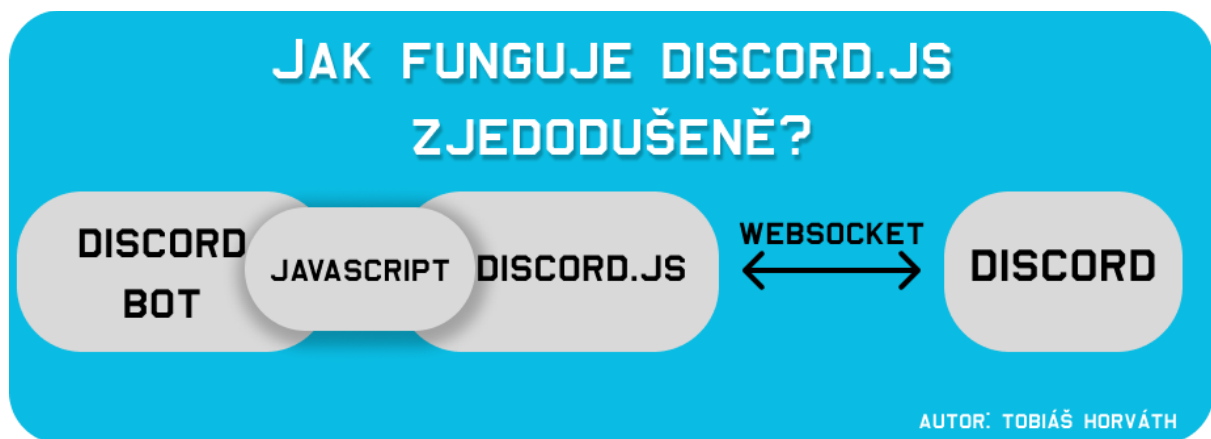
Discord bot vyvinutý pomocí knihovny Discord.js je softwarový robot, který je schopný komunikovat s Discordem, což je komunikační platforma primárně pro hráče počítačových her. Tento bot je implementován pomocí JavaScriptu a vyžaduje, aby byly na počítači nainstalovány Node.js a npm (node package manager).

Komunikace mezi botem a Discordem probíhá prostřednictvím protokolu WebSocket, což je protokol umožňující plně duplexní (obousměrné) spojení mezi klientem a serverem přes síť. Discord.js knihovna poskytuje API pro snadnou práci s tímto protokolem a umožňuje vývojářům psát příkazy pro ovládání bota.

V rámci vývoje bota jsou k dispozici různé události a funkce dostupné v Discord.js knihovně, které umožňují reagovat na události, jako je příchod nové zprávy nebo připojení nového uživatele k serveru, a také umožňují odesílat zprávy a manipulovat s kanály a rolemi na serveru.

### 2.2.1 Co je to API?

API (Application Programming Interface – rozhraní pro programování aplikací) je soubor pravidel a definic, které umožňují jedné aplikaci komunikovat s jinou aplikací nebo službou. API obvykle obsahuje dokumentaci, která popisuje, jakým způsobem s ním lze pracovat u knihovny Discord.js. Dokumentaci můžete najít na webové adrese <https://discord.js.org/>, a zpravidla dokumentace obsahuje také druh specifikace, například v podobě kódu ve formátu XML nebo JSON, který je používán pro vytváření a odesílání požadavků a přijímání odpovědí.



Obrázek 7 discord.js

API se často používají pro přístup k různým službám nebo datům, které jsou dostupné na internetu, například pro získávání informací z webových stránek nebo pro připojení se k službám typu Google Maps nebo Twitter, v tomto případě ke službě Discord Developer Portal. API jsou také často používány pro integraci různých aplikací a služeb mezi sebou. Právě knihovna Discord.js pomáhá developerům komunikovat s Discord API mnohem jednodušším



způsobem, než kdyby developeři rovnou pracovali přímo s Discord API, která je komplikovanější z důvodu potřeby znalosti protokolu WebSocket. Výhodou však je, že můžete využívat novinky okamžitě od momentu jejich vydání a nemusíte čekat na novou verzi knihovny.

## 2.2.2 Proč Discord využívá WebSocket a k čemu to je?

WebSocket je protokol pro založení plně duplexního spojení mezi klientem a serverem přes síť. Je navržen tak, aby umožňoval komunikaci mezi stranami bez nutnosti iniciovat požadavek od klienta. To umožňuje klientovi a serveru odesílat a přijímat zprávy v reálném čase. WebSocket se spojuje se serverem pomocí připojení HTTP a poté se přepne na WebSocket protokol. Po spojení je otevřeno plně duplexní spojení mezi klientem a serverem, které je udržováno po celou dobu trvání spojení. WebSocket je často používán pro aplikace, které vyžadují rychlou odezvu, nebo pro aplikace, které potřebují odesílat a přijímat zprávy v reálném čase. Také je navržen tak, aby byl co nejmenší a aby nezatěžoval síť příliš velkým množstvím dat. Proto je tento protokol nejvíce využíván službami jako jsou online hry nebo aplikace pro chatování, např. právě Discord.

## 2.2.3 Databáze a ukládání dat o uživateli

PetBot využívá pro ukládání dat o uživateli nebo jiných informacích skladovaných pro pozdější využití databázový systém MongoDB. Bot ukládá mnoho informací o uživateli, ale žádné osobní údaje. Ukládá např. jejich údaje o herní postavě (level nebo ID vlastněných mazlíčků).

MongoDB je databázový systém s otevřeným zdrojovým kódem, který je tzv. "NoSQL". To znamená, že se nesnaží používat SQL jako svůj hlavní jazyk pro práci s daty, ale místo toho používá jiný typ databázového jazyka založeného na dokumentech. MongoDB je známý pro svoji schopnost pracovat s velkými množstvími dat a pro svoji flexibilitu při práci s různorodými typy dat. Místo tabulek s řádky a sloupci používá MongoDB k ukládání dat do dokumentů ve formátu JSON, což umožňuje snadné ukládání a získávání dat bez nutnosti definovat předem schéma tabulky. MongoDB je často používán pro webové aplikace a je kompatibilní s mnoha jazyky včetně JavaScriptu, Pythonu a C#. Je také schopný fungovat v distribuovaném prostředí s vysokou dostupností a lze jej použít pro ukládání velkého množství dat bez nutnosti ručně spravovat výkon databáze.

Veškerí mazlíčci jsou uloženi v další databázi se všemi informacemi o nich, od barvy, jména i po státy a jejich generované ID, které zde funguje jako primární klíč, stejně tak u uživatelů je jejich primární klíč originální Discord ID. Primární klíč je unikátní identifikátor, který se používá v databázi pro jednoznačné identifikování řádku v tabulce. Každá tabulka v databázi by měla mít primární klíč, který je používán pro vyhledávání a úpravu záznamů v tabulce. Bez

primárních klíčů by se mohlo stát, že se objeví 2 identičtí adoptovaní mazlíčci se stejnými jmény a údaji, a pak by mohl nastat problém v kódu při rozpoznávání těchto dvou mazlíčků.

## 2.2.4 Struktura příkazů bota

Příkaz pro Discord bota se obvykle skládá ze tří částí: listener eventu, podmínky a reakce.

Listener eventu je kód, který naslouchá určité události nebo akci, která se může vyskytnout v rámci Discordu. Například to může být event "message", což znamená, že bot naslouchá příchodu nové zprávy v chatu. Může to být také event "guildMemberAdd", což znamená, že bot naslouchá příchodu nového uživatele do serveru.

Podmínky jsou kód, který určuje, zda má být reakce bota vyvolána, nebo ne. Podmínkou může být, že reakce bota se spustí pouze v případě, že zpráva obsahuje určité slovo nebo frázi. Dalším příkladem je, že reakce bota se spustí pouze, pokud zpráva pochází od určitého uživatele nebo je odeslána do určitého kanálu. Tím se mohou vytvářet konkrétní příkazy, které naslouchají konkrétním slovům.

Reakce je kód, který se vykoná, pokud jsou splněny všechny podmínky. Reakce může být například odpověď bota na zprávu nebo přepsání dat v databázi, např. zvýšení hodnoty hladu u mazlíčka.

```
module.exports = {  
  command: "wash",  
  execute: async function(message, client){  
    var selectedPet = await modules.selectOwnedPet(message)  
    if(!selectedPet) return
```

Zde kód čeká, než si uživatel vybere jakého mazlíčka chce umýt.

```
    if(selectedPet.pet_id == "fish"){  
      message.reply(await modules.addPetImageToEmbed(selectedPet, fishes)).  
      return  
    }  
  }
```

Kontrola zda mazlíček není ryba, jelikož ta umýt nejde.

```

        if(selectedPet.asleep){
            message.reply(await modules.addPetImage(selectedPet.petStats, selectedPet.petStats.price_for_wash, selectedPet.petStats.price_for_sleep).setDescription("Type `"+ config.bot.prefix + "sleep` to wake him up"))
        }
        return
    }
}

```

Kontrola zda mazlíček nespí.

```

        .calculatePoints(message.author.id, selectedPet.petStats) / selectedPet.petStats.price_for_sleep
        var points = await modules.calculatePoints(message.author.id, selectedPet.petStats) / selectedPet.petStats.price_for_sleep
        price = Math.ceil(( / 100 * selectedPet.petStats.price_for_wash) * 100)
        if(price > points) return discordEmbeds.notEnoughFun(message.reply("You can't wash your pet with that many points"))
        return discordEmbeds.warning("This pet is sleeping")
    }
    (points,

```

Kód vypočítá cenu umytí mazlíčka ze základní ceny a jeho věku.

```

        const pet = await modules.getPet(message.author.id)
        if(pet) return discordEmbeds.warning("This pet is already owned")
        modules.mongodb.collections.points.insertOne({user_id: message.author.id, xp: selectedPet.xp, price_for_wash: selectedPet.petStats.price_for_wash, price_for_sleep: selectedPet.petStats.price_for_sleep})
        modules.mongodb.collections.ownedPets.updateOne({user_id: message.author.id, pet_id: selectedPet.petId}, {price_for_wash: selectedPet.petStats.price_for_wash, price_for_sleep: selectedPet.petStats.price_for_sleep}, {upsert: true})
    }
}

```

```
price))
```

```
message.author.id, count: price * -  
)  
_id: selectedPet._id}, {$set:  
{hygiene:  
max_hygiene -  
selectedPet.hygiene) / 10}})
```

Zde probíhá přepis dat v databázi. Snížení Pet Coinů uživatele a zvýšení hygieny a levelu mazlíčka.

```
selectedPet = await modules.getOwnedPet(selectedPet._id)
```

```
message.reply(await modules.addPetImageToEmbed(discordEmbeds.default("You have washed " +  
selectedPet.name).setDescription("It cost you " + price + " " + config.bot.pointsEmoji)  
).addField(":100: ", selectedPet.level, true), "Level",  
`${selectedPet.level}`\n${modules.createProgressBar(selectedPet.level_progress, 100)})  
).addField(":shower: ", selectedPet.hygiene, true), "Hygiene",  
modules.createProgressBar(selectedPet.hygiene, selectedPet.max_hygiene)), selectedPet)  
)  
}
```

} Na konci kód vypíše uživateli, kolik ho umytí stálo.

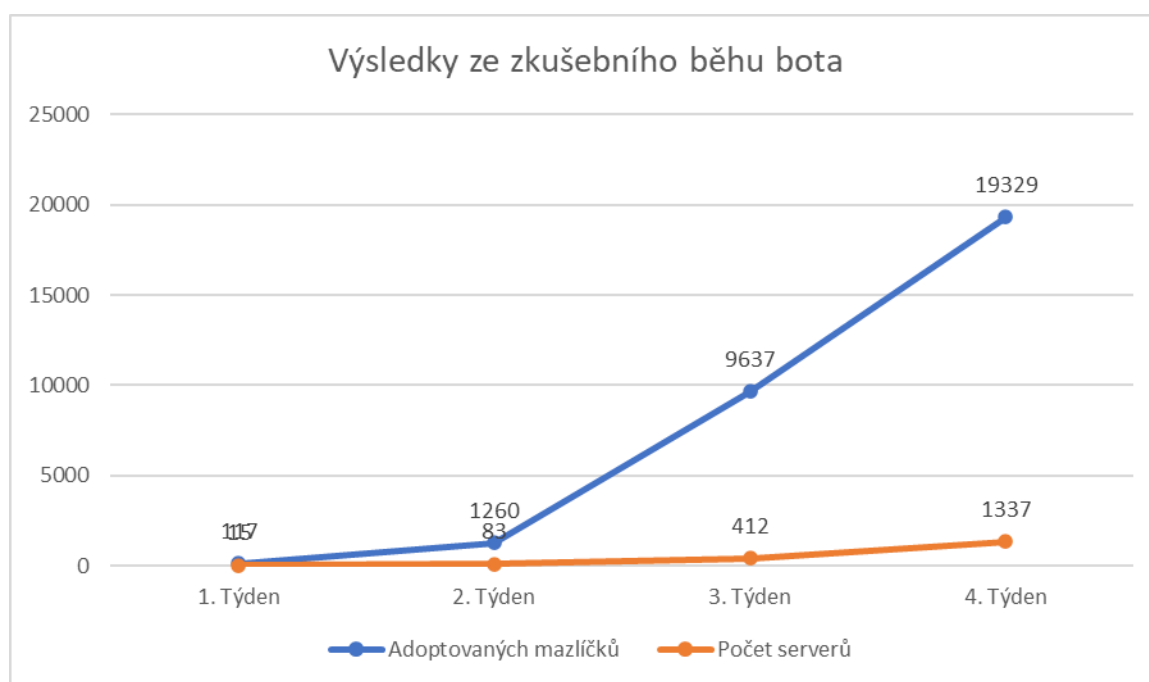
## Závěr

Cílem mé odborné práce bylo vytvořit pomocí Discord bota, díky kterému se uživatelé mohou starat o své vlastní mazlíčky. Během vytváření bota jsem musel řešit mnoho faktorů při vybírání knihoven, databází a další nástrojů potřebných pro běh Discord bota. Snažil jsem se vytvořit bota tak, aby uživatelé nemohli získat ty nejlepší mazlíčky ihned v prvních dnech a hra je tak vybízela k dalšímu hraní a k posouvání.

Při programování jsem musel řešit i poměrně časté aktualizace knihovny Discord.js, kvůli kterým jsem musel bota několikrát aktualizovat, jelikož by bot jinak nemohl běžet na nejnovější a nejbezpečnějších systémech. Přes tyto překážky si myslím, že jsem vytvořil kvalitního Discord bota dle mých veškerých představ. Během vytváření bota jsem si zopakoval mnoho schopností, nabyl pár tvrdých dovedností a zlepšil i některé své slabší stránky.

Momentálně je bot nasazen jako minihra již na několika serverech. Každý den dostávám zpětné reakce uživatelů a snažím se bota upravovat na základě jejich požadavků. Bot má již i vlastní komunitu každodenních hráčů, kteří si své mazlíčky vyměňují a získávají díky tomu další.

Tabulka 1 Graf výsledků ze zkušebního běhu bota.



Graf znázorňuje měsíční testovací běh bota během, kterého byl bot veřejně dostupný k pozvání na vlastní servery a spuštěn pro lidi, aby si mohli vyzkoušet starání o své digitální mazlíčky. Díky této popularitě bot získal od Discordu ověření, během kterého byl otestován pracovníky z Discordu. Díky tomuto ověření bot získal mnohem více možností jak se rozrůstat a uživatelé se už nemusí bát, že by se jednalo o škodlivého bota. Dle mé prognózy by se měl bot při jeho znovuspuštění do měsíce dostat na minimálně 10.000 serverů.

V první kapitole mé teoretické části popisují další možné komunikační platformy pro vytváření botů a jejich obecné výhody a nevýhody. Do těchto komunikačních platform patří Discord, Telegram, Facebook a Twitter. Na konci kapitoly zmiňuji důvody, proč jsem se nakonec rozhodl pro komunikační platformu Discord.

V druhé kapitole představuji možné knihovny pro programování botů na platformu Discord. Pojednávám o knihovnách Discord.js, Discord.py, Eris, JDA a discordrb. Zmiňuji zde jazyky, které se u těchto knihoven používají, velikost komunity a jak dobře mají popsané dokumentace. Na konci kapitoly popisují a odůvodňují svůj výběr.

Třetí kapitola se dělá na dvě části. První pojednává o potřebných systémech, datech a informacích pro vytvoření Discord bota a druhá pojednává o tvorbě Discord bota dle vlastní konfigurace.

V první kapitole mé aplikační části pojednávám o základních příkazech Discord bota a prvních momentech z pohledu uživatele, tedy co vše může dělat a jak může začít s chováním svého digitálního mazlíčka.

Druhá kapitola se dělá na čtyři části a pojednává o botovi z IT hlediska. První část zodpovídá otázku, co je to API a jak jí bot využívá. Druhá část jedná o protokolu WebSocket, který Discord využívá a jak s ním pracuje vybraná knihovna Discord.js. Třetí část pojednává o databázi, kterou bot využívá pro ukládání dat o uživateli, mazlíčcích a veškerých dalších uchovávaných informacích. Hlavní náplň poslední části je struktura příkazu Discord bota, zbytek kapitoly popisuje přímo kód příkazu „/wash“ z bota.

## Použitá literatura

*Discord Developer Portal* [online]. San Francisco: Discord, 2022 [cit. 2023-01-08]. Dostupné z: <https://discord.com/developers/>

*Dev Docs* [online]. London: Dev Docs, 2021 [cit. 2023-01-08]. Dostupné z: <https://devdocs.io/javascript/>

*Discord.js* [online]. New York: Discord.js, 2022 [cit. 2023-01-08]. Dostupné z: <https://discord.js.org/#/>

*Discordpy Readthedocs* [online]. London: Read the docs, 2021 [cit. 2023-01-08]. Dostupné z: <https://discordpy.readthedocs.io/en/stable/>

*Abal Moe* [online]. Amerika: Abal.moe, 2021 [cit. 2023-01-08]. Dostupné z: <https://abal.moe/Eris/>

MinnDevelopment. *Github.com* [online]. New York: Github.com, 2022 [cit. 2023-01-08]. Dostupné z: <https://github.com/DV8FromTheWorld/JDA>

Birdie0. *Github.com* [online]. New York: Github.com, 2021 [cit. 2023-01-08]. Dostupné z: <https://github.com/shardlab/discordrb>

*Twitter.com* [online]. New York: Twitter.com, 2023 [cit. 2023-01-08]. Dostupné z: <https://developer.twitter.com/en/docs>

*Facebook.com* [online]. San Francisco: Facebook.com, 2023 [cit. 2023-01-08]. Dostupné z: <https://developers.facebook.com/>

*Telegram.org* [online]. London: Telegram.org, 2022 [cit. 2023-01-08]. Dostupné z: <https://core.telegram.org/>

*Readthedocs.io Websocket* [online]. London: Readthedocs.io, 2021 [cit. 2023-01-08]. Dostupné z: <https://websockets.readthedocs.io/en/stable/>

*MongoDB Dokumentace* [online]. London: MongoDB.com, 2020 [cit. 2023-01-08]. Dostupné z: <https://www.mongodb.com/docs/>

## Seznam Obrázků

<a href="#">Obrázek 1 Příkaz Adopt</a> .....	10
<a href="#">Obrázek 2 Příkaz Mypet</a> .....	11
<a href="#">Obrázek 3 Příkaz Myself</a> .....	11
<a href="#">Obrázek 4 Speciální dospělý pes</a> .....	12
<a href="#">Obrázek 5 Dospělý pes</a> .....	12
<a href="#">Obrázek 6 Mazlíci</a> .....	13
<a href="#">Obrázek 7 discord.js</a> .....	14