DISCORD BOT

A SYP Project

4. OKTOBER 2018

TISCHLER, TRINKL, STRUTZENBERGER, PUNZ
HTL-Leonding

Content

1. Introduction

Every bot has different functions so the problem is that one has to add many bots in order to use the functions one wants. The following problem is that every bot has different exclamations and as said above different functions. So in order one can use the functions one wants the user has to remind every exclamation of every bot on his server. Our project idea is to combine the most needed functions of the community in one single bot with one unified exclamation. The project leader is Trinkl Philip, and the others are Strutzenberger Chiara, Tischler Fabian and Punz Flo.

2. Initial Situation

Right now, there are many bots with only a few functions. Our idea is to make a big bot with many functions to make it easier for the users. It is always a bit annoying that there are many bots with different exclamations and many functions are on the server more than once due to the big number of bots there are. With our bot we try to take most (the most important) functions from the well-known bots and put them together on our bot.

The general aim of discord bots is to give the user much more functions and make it easier for the admins to administrate the server. There are also some bots which are made for specific games. They give a few guidelines what to do and give general information about the game, like the last games of a player. Overall it makes discord much more user-friendly.

For example, on our server (WilheringerInferno), we have three bots that can play music. We have them because every bot of them has a unique function we wanted to have. That is not what we want, so we got the idea of our big bot.

We are very determined because we wanted to make a discord bot for a long time but we never had the time to do it. We think also that our programming skills are more than enough for that (C#).

Example:

Bot1:

!!play <YouTube link>

!!pet

++++++++++++++++++++++

Bot2:

+play <YouTube link>

+userprofile

Bot one and two have different exclamations and different and same functions. In this example we combine all the function and unify the exclamation in our bot.

➔➔➔➔➔➔➔➔➔➔➔

Our Bot:

!play <YouTube link>

!pet

!userprofile

3. General Conditions and Constraints

General Conditions

Know-how in discord is required for making bots. We have to research the whole C# library from discord plus some more specific libraries from discord like discord voice. The users don't need much know-how in discord to use the bot. In the most cases a bot provides a help function. An account is needed to publish a discord bot. If you enter the website from discord there is a tab called developer portal. You need to login with your account and then you can add a bot. The bot is distributed by the official discord bot website (https://discordbots.org/). If one sets their bot public, it automatically can be found on this website.

Framework Conditions

For a discord bot we do not need any money, so our budget situation is very good.

As said above, we have a very good knowledge of C# (in this language we want to write our bot) and we are not that bad in java and C as well. We do not know a lot about discord bots (programming) so we hope that we can improve our skills in this area.

Our Deadline is the end of the school year, so we think it is not a problem to finish until then.

Technical Conditions

We want to program in Visual Studio (C#), which is sponsored by our great school (HTBLA Leonding). We work on our Laptops (Windows 10) and share our data on GitHub and Discord. The API we use is Discord.NET, which supports Visual Studio and .NET Core SDK. Its an unofficial .NET API Wrapper for the Discord client (http://discordapp.com). Discord Bots can be written for example in: C#, Java, Go, Lua, NodeJS, PHP, Python, Ruby. The program size for discord bots is not limited. You host them locally or from a server, so you do not have to upload them. The size is only limited by the size of the hard disk for example.

4. Project Objectives and System Concepts

Vision of our new System:

We want to make a Discord bot which implements most of the features that are already implemented in other big bots and some completely new. So now we want to unify all exclamations from every bot by implementing the features in our bot so that we can use all functions with one exclamation. Our bot provides functions like “!play”, “!pet” or “!userprofiles”. For example the “!play” function provides you with the opportunity to play music. The command has to be written in a text channel. One has to add to the “!play” command for example a YouTube link. If the YouTube link is valid the bot will join the voice channel and plays the requested YouTube video.

The functions we want to implement are:

- Play Music: It is a function which gives the user the opportunity to play a requested YouTube Video
- Memes: It should send a random meme from our data base to a channel
- User Profiles: With the user profiles we want to implement a function which displays the connections (Steam, Spotify, Riot, …), the level, the coins and the pet of a user. One can raise his level with typing messages in text channels.
- Pet: This should be a function which is similar to a Tamagotchi. One can pick a pet, give him a name, play with it or feed it.
- Coins: With the coins one can buy roles. Coins are obtained by minigames and level-ups.
- Roles: The roles are displayed which one can buy. They all give a user more rights.
- Minigames: We want to implement several minigames with which one can obtain coins.
- White/Blacklist: This is a restriction who can use the bot. Users who are on the Whitelist can use every function the bot is providing. Users on the blacklist are restricted which means they aren’t allowed to use any functions of the bot at all.

5. Opportunities and Risks

Opportunities:

We are able to prevent that the user have to use too many exclamations and we are able to combine many features in one single bot. The number of users is not displayed it only tells one on how many servers the bot got invited. We expect a minimum of 7 server. Bots are usually free but we have the opportunity to implement a donate function.

Risks:

It can happen that the bot gets too many requests to handle at the same time, so it crashes or starts to lag. The performance of the bot is depending on the internet connectivity of the server and the pc where the bot is hosted. It can happen that the community does not find our bot because when a member of the community first enters the website the bots with the most uploads are displayed at the top and the new bots are displayed at the end. So new bots often do not get a chance. The community could also not accept our bot because are not as flexible as the ones from the well-known bots (for example “!play” can play YouTube or Soundcloud).
We set us too many goals.

6. Planning

Project milestones:

- Project proposal 19.10.18
- Get the bot running (login/out) 9.11.18
- Major functions (music, memes, user profiles) 20.1.19
- Test the functions /correct the functions 1.2.19
- More functions (pet, steam, coins, roles, minigames, white/blacklist) 31.5.19
- Test the functions/correct the functions 20.6.19

Project lead:

Phillip Trinkl (control the progress, set/change the goals/programming functions)

Programming lead:

Florian Punz (connect all the functions to the bot/programming functions)

Programmer:

Fabian Tischler (programming functions)

Chiara Strutzenberger (programming functions)

Resources:

Licenses and Servers are not needed (the bot is locally hosted, discord bots are free to make, the only license we need is provided from our school (Visual Studio))

Project Start: 27.9.18

Project End: 5.7.19

First Prototype date: 1.2.19

Implementation work start date: -