Python Project Proposal

13006107 Introduction to Computers and Programming

Software Engineering

Faculty of Engineering, KMITL

Project Name:

Discord Music Bot (The Bot)

By

Anucha Cheewachanon 64011338

Contents

Introduction and Motivation

Following YouTube's agreement with Discord to remove the majority of bots because they directly affect YouTube's revenue, most servers now have almost no music bots, as the most popular bots are being phased out. For example, Rythm, Fredboat, Groovy, and so on. By relaying music directly into a voice channel, these bots have an impact on YouTube's ad revenue. Some bots even have the option to upgrade to a premium version by paying the bot's creator rather than YouTube.

This has an effect on the servers I moderate as well. Discord removed the bots we were using. On the public server, one of the solutions we used was to use another bot that had survived, in this case, Manybaht. Manybaht does not have a premium option and is not well-known internationally. As a result, Manybaht is Thailand's most popular music bot. I, for one, have a bias against the bot and have chosen the alternative path of self-hosting. I repurposed the bot token I used for the Minecraft chat system back in April to make a music bot. In my private server, I used [jagrosh's JMusicBot] as the engine for my bot, "Sierokarte." It worked flawlessly with almost no errors.

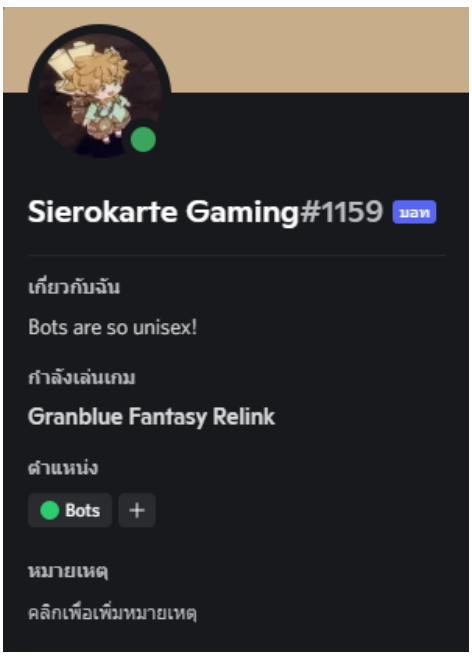


Figure 1: Manybaht's Bot profile on Tomkai, the public server

Figure 2: Sierokarte's Bot Profile, running JMusicBot on LMAO (sic), the private server

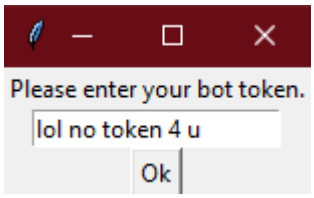
I had an idea while playing around with Python and its modules: I want to write a music bot in Python to replace JMusicBot, which Sierokarte currently uses.

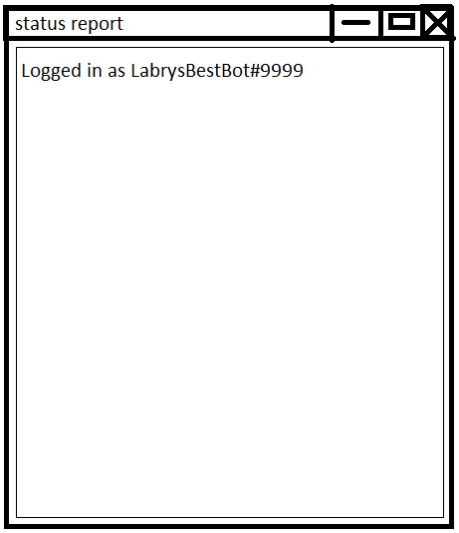
It turns out that discordpy, a python module for creating a discord bot, comes into play. Discordpy can read the music file and play it into a voice channel using youtube-dl and ffmpeg, according to further reading.

Customizable command and config file that saves various configurations (Bot token, command, etc.) and using Tkinter's Entry for the status report are the features I want. I also decided to add a flag to the remote server that can be used to report the status to the command line instead, which is ideal for SSH.

Rough GUI sketches:

If the config file is incorrect or not presen, the program displays a Tkinter dialogue prompting the user to enter the bot's token. The status report screen consists of a single uneditable entry widget what appears after the token has been added (or there is an existing token in the config file). It should resemble a small notepad window that can't be edited.





The bot commands include, but are not limited to, play, skip, and other basic functions. The bot will respond with the command help via DM if the client requests assistance. The bot will instead paste it into the same channel if a specific flag is set. (The image contains a placeholder string and a placeholder command to demonstrate the feature.)