

Project report: gloobot

Specifications:

- **Objective:**

To create a simple Bot for the communication platform Discord for people to listen to music and send gifs on a discord server using python and Discords module, Discord.py.

- **Description:**

The Discord bot will be named “gloobot” it derived from the developer online game tag. It is designed so that the Bot could play music on the server’s voice channel and send gifs on the server’s text channels. by simply inputting a YouTube URL link to the commands the bot will play music for the users to listen to. And the users could input commands so the bot will send gifs according to the command.

Files involved:

main.py and commands_cog.py

Modules and other requirements:

- Discord.py – a module by Discord to create bots using Python
- Yt_dlp - a module for downloading YouTube video
- FFMpeg – the mp3 player
- os
- Random(choice)

Main.py:

```
main.py M x  commands_cog.py U
E: > werk > Algorithm and programming > Exercise(LEQ) > gloobot-fp > main.py > ...
1  from random import choice
2  import discord
3  from discord.ext import commands, tasks
4
5  from commands_cog import commands_cog #importing the cog
6
7  client = commands.Bot(command_prefix="-", intents = discord.Intents.all()) #initiate the bot command prefix to "-" and letting the user to use every commands
8  status = ["Studying!", "HOW LONG IS THIS GOING TO TAKE!"] #sets of status
9
10 client.add_cog(commands_cog(client)) #registering the class with the Bot
11
12 @client.event #when the bot is online it will activate the change_status function and print out "Bot is online"
13 async def on_ready():
14     change_status.start()
15     print("Bot is online!")
16
17 @tasks.loop(seconds=20) #changes the status the bot every 20 seconds by random
18 async def change_status():
19     await client.change_presence(activity=discord.Game(choice(status)))
20
21 TOKEN = "OTIzOTMs0Tjc1MTQyNzg3MDcy.YCXM4DQ.TAtvMhYw9pbSMB0tNkXoxA_SNE"
22 client.run(TOKEN)#the token is used for connecting the Bot (it couldn't be posted publically since token are basically a password for the bot account)
```

Main.py main purpose is to become the Driver for the command.py file.

```
client = commands.Bot(command_prefix="-", intents = discord.Intents.all()) #initiate the bot command prefix to "-" and letting the user to use every commands
status = ["Studying!", "HOW LONG IS THIS GOING TO TAKE!"] #sets of status
```

The client variable is made to initiate the Bot and setting the Bot command prefix to “-” and also letting the user to use all of commands from the module.

```
15 @client.event #when the bot is online it will activate the change_status function and print out "Bot is online"
16 async def on_ready():
17     change_status.start()
18     print("Bot is online!")
19
20 @tasks.loop(seconds=20) #changes the status the bot every 20 seconds by random
21 async def change_status():
22     await client.change_presence(activity=discord.Game(choice(status)))
23
```

The on_ready function is a function that let the user know when the Bot is online It will print out “Bot is online!”. And it will start the function change_status, the function change_status will display the

status written on the status variable. The status will change every 20 seconds.

Commands_cog.py:

Commands.py is the class file, it has the commands code inside it.

```
YDL_OPTION = {
    'format': "bestaudio",
    'postprocessors': [{
        'key': "FFmpegExtractAudio",
        'preferredcodec': "mp3",
        'preferredquality' : "192"}],
    } #the music format
```

YDL_OPTION is the option or configuration for the downloaded mp3 file. 'format' is so that it downloads the best audio quality for 'key' it so that FFmpeg extract the audio, 'prefferedcodec' so that when yt_dlp download a video from YouTube it will download it to the mp3 format.

```
class commands(commands.Cog):
    def __init__(self, client):
        self.bot = client

    @commands.command(name = "join", help = "This command makes the gloobot join a channel") #commands to make the bot join a discord voice channel
    async def join(self, ctx):
        if ctx.author.voice is None:
            await ctx.send("Bro, you're not in the channel")
            voice_channel = ctx.author.voice.channel
        if ctx.voice_client is None:
            await voice_channel.connect()
            await ctx.send(f"***joining** `{voice_channel}`")
        else:
            await ctx.voice_client.move_to(voice_channel)
            await ctx.send(f"***we movin to** `{voice_channel}`")
```

The Join function is a command for the Bot to join the Discord voice channels. If the users inputting the join command is not inside a voice channel the Bot will send "Bro, you're not in the channel" telling the user to get into the channel first. If the user is already inside a voice channel, then the Bot will join the Voice channel and will send "joining (the name of the channel)". And if the Bot is already in an existing

voice channel but the user is on another voice channel the Bot will move to the user's new voice channel and will output "We movin to (the name of the channel)"

```
@commands.command(name = "leave", help = "This command makes gloobot to leave the channel") #commands to make the bot disconnect from a discord voice channel
async def leave(self, ctx):
    voice_channel = ctx.author.voice.channel
    await ctx.voice_client.disconnect()
    await ctx.send(f"**Adios** `{voice_channel}`")
```

The leave function is command made so when the user input the command the bot will leave the Voice channel and send "Adios (the name of the channel) to the text channel

```
@commands.command(name = "play", help = "to use this command type -play *url*") #commands to make the bot play music
async def play(self, ctx, url):
    song_there = os.path.isfile("song.mp3")
    try:
        if song_there:
            os.remove("song.mp3")
    except PermissionError: #since the Bot doesn't have queue system the user need to wait for the music to finish or use the stop command
        await ctx.send("Wait for the current playing music to end or use the 'stop' command")
        return

    vc = ctx.voice_client

    with yt_dlp.YoutubeDL(YDL_OPTION) as ydl: #downloading the music from youtube to the local machine
        ydl.download([url])
    for file in os.listdir("./"): #changes the music file name to "song.mp3"
        if file.endswith(".mp3"):
            os.rename(file, "song.mp3")
            source = discord.FFmpegPCMAudio("song.mp3")

    vc.play(source)
    await ctx.send(f"**now playing** `{url}`")
```

The function play is a command made so when the user input the command "-play (YouTube URL)" it will play the music inside the voice channel. How It works is that the Bot will download the video to the local machine and after it done downloading the video it will replace the name to "song.mp3". after the song being renamed the music will be the source of the audio being played in the voice channel. Then the source audio is played inside the voice channel and sending "now playing (YouTube URL)" to the text channel.

```
@commands.command(name = "pause", help = "This command pause the music gloobot is playing") #commands to make the bot pause the music
async def pause(self, ctx):
    ctx.voice_client.pause()
    await ctx.send("Paused music")
```

The pause function is a command to let the user to pause the music that is currently playing after it paused the music it will send "Paused music" to the text channel

```
@commands.command(name = "resume", help = "This command resumed the song that are being paused") #commands to make the bot resume the music
async def resume(self, ctx):
    ctx.voice_client.resume()
    await ctx.send("Resume music")
```

Just like the pause function the resume function is a command to let the user to resume the paused music after it resumed the music it will send "Resume music"

```
@commands.command(name = "stop", help = "stop the music that is playing") #commands to make the bot stop the music
async def stop(self, ctx):
    ctx.voice_client.stop()
    await ctx.send("Stopped music")
```

The stop function is command to let the user stop the music that are currently playing since there is no queueing system inside the bot. if the user wanted to change the music the user could use the stop command or wait until the music is finished.

```
@commands.command(name = "happy", help = "use this command to send a happy gif") #the command will send an happy gif :)
async def happy(self, ctx):
    gif = [
        "https://tenor.com/view/yay-anime-girl-kawaii-gif-18081573",
        "https://tenor.com/view/black-clover-anime-guy-chico-asta-gif-19791391",
        "https://tenor.com/view/anime-excited-happy-smile-gif-15060821",
        "https://tenor.com/view/adachi-anime-happy-happy-gif-19233583",
        "https://tenor.com/view/anime-happy-anime-excited-gif-19679255",
        "https://tenor.com/view/anime-yay-happy-excited-yeah-gif-9528804",
        "https://tenor.com/view/anime-happy-excited-gif-13451198",
        "https://tenor.com/view/chika-chika-dance-anime-anime-dance-dance-gif-13973731",
        "https://tenor.com/view/chi-chis-sweet-home-comfy-kitty-cute-gif-16140552",
        "https://tenor.com/view/anime-happy-cute-excited-gif-12057651",
        "https://tenor.com/view/happy-japanese-anime-excited-gif-9596035",
        "https://tenor.com/view/komi-komi-san-komi-interested-komi-neko-mimi-komi-cat-ear-gif-22007033",
        "https://tenor.com/view/komi-san-komi-shouko-komi-shouko-komi-cant-communicate-gif-21525381",
    ]
    await ctx.send("someone is feeling good")
    await ctx.send(choice(gif))
```

The function happy is a command made so when the user input the command it will output a happy looking gif and sending the message “someone is feeling good” to the text channel. By using choice from the random module, the bot could send out one of the gifs randomly.

```
@commands.command(name = "sleepy", help = "use this command if you feel sleepy") #sleepy gif cause i need some sleep
async def sleepy(self, ctx):
    gif = [
        "https://tenor.com/view/sleepy-sleep-anime-anime-sleep-anime-sleepy-gif-24142121",
        "https://tenor.com/view/willcore-kon-anime-girl-sleepy-gif-24035077",
        "https://tenor.com/view/sleepy-nichijou-tired-yawn-wipe-eyes-gif-16309858",
        "https://tenor.com/view/kanna-kawai-tired-sleep-sleepy-gif-15961742",
        "https://tenor.com/view/tired-anime-kawaii-blue-hair-sigh-gif-17714309",
        "https://tenor.com/view/yawn-tired-anime-manga-japanese-manga-gif-9525859",
        "https://tenor.com/view/sleeping-anime-my-hero-academia-eraserhead-shota-aizawa-gif-16604988",
        "https://tenor.com/view/tired-kon-anime-yawn-sleepy-gif-17415905",
        "https://tenor.com/view/gintoki-no-sleep-sleepy-gif-12592776",
        "https://tenor.com/view/anime-girl-sleepy-gif-22569001",
        "https://tenor.com/view/umaru-sleeping-sleep-anime-gif-12007584",
        "https://tenor.com/view/nadeshiko-laid-back-camp-anime-sleepy-gif-12003890"
    ]
    responses = ["***YAWN*** i need to slee-", "*stretch-strech*"]
    await ctx.send(choice(responses))
    await ctx.send(choice(gif))
```

The sleepy function is a command made so when the user input command it will output a sleepy anime character gifs and 2 responses which is “**YAWN** I need to slee-” and “*stretch-strech*”. Like the happy function it uses the choice from random module to randomize the output of the gif and the response.

```
@commands.command(name = "sad", help = "sends a sad gif T_T") #sends an sad anime gif T_T
async def sad(self, ctx):
    gif = [
        "https://tenor.com/view/anime-crying-sad-gif-14210687",
        "https://tenor.com/view/crying-cute-anime-sad-tears-gif-16038248",
        "https://tenor.com/view/sorry-crying-anime-sad-gif-15171171",
        "https://tenor.com/view/trash-disappointed-no-sad-bye-gif-5005980",
        "https://tenor.com/view/anime-cry-sad-gif-14080503",
        "https://tenor.com/view/anime-girl-sad-cry-tears-gif-17100832",
        "https://tenor.com/view/llorar1-cry-sad-tears-anime-gif-5648908",
        "https://tenor.com/view/anime-gif-19105479",
        "https://tenor.com/view/aqua-aqua-crying-crying-gif-21481711",
        "https://tenor.com/view/deku-cry-tears-anime-izuku-midoriya-gif-14926648",
        "https://tenor.com/view/hunter-x-hunter-gon-freecs-sad-crying-tears-gif-16729297",
        "https://tenor.com/view/sorry-crying-anime-sad-the-demon-girl-next-door-gif-15005984",
        "https://tenor.com/view/anime-cry-wataten-crying-gif-13356071",
        "https://tenor.com/view/anime-hitori-bocchi-no-marumaru-seikatsu-crying-sad-gif-14016926",
        "https://tenor.com/view/komi-komi-san-komi-cry-komi-san-cry-gif-24002547",
        "https://tenor.com/view/anime-sad-anime-pout-anime-sorry-horimya-hori-gif-20710638",
    ]
    responses = ["someone is not feeling so good...", "***pat** *there there...*"]
    await ctx.send(choice(responses))
    await ctx.send(choice(gif))
```

The sad function works the same as the sleepy and happy function when the user input the command it will output a random gif and response to the text channel.

```
@commands.command(name = "pout" , help = "send a pouting gif")
async def pout(self, ctx):
    gif = [
        "https://tenor.com/view/komi-komi-san-pout-anime-anime-pout-gif-23453143",
        "https://tenor.com/view/anime-raphalia-mad-pouty-gif-16985978",
        "https://tenor.com/view/pout-anime-pout-sad-cute-gif-17524621",
        "https://tenor.com/view/pout-hmph-anime-girl-kawaii-gif-17549065",
        "https://tenor.com/view/anime-angry-mad-gif-14108774",
        "https://tenor.com/view/wataten-watashi-ni-tenshi-ga-majorita-pout-hinata-anime-gif-16058457",
        "https://tenor.com/view/nagatoro-hayase-nagatoro-nagatoro-pout-pout-pouting-gif-22127015",
        "https://tenor.com/view/anime-pouting-grumpy-gif-13451362",
        "https://tenor.com/view/raphalia-anime-pouting-gif-14210688",
        "https://tenor.com/view/anime-gif-12942766",
        "https://tenor.com/view/answer-me-the-quintessential-quintuplets-5toubun-no-hanayome-gif-21661590",
    ]
    await ctx.send("oh uh someone is mad...")
    await ctx.send(choice(gif))
```

The pout function work also the same way as the other three functions when the user input the pout command it will output a random pouting anime gif and the text “oh uh someone is mad...” to the text channel

Demo pictures:

