|  |
| --- |
| **Project Showcase** |

1. <https://www.linkedin.com/in/henri-georges-demanou-tchamdjou>
2. <https://github.com/henri500/Python-Discord-bot/wiki>
3. <http://hilite.me/> ***format code***

**Introduction:**

This aim of this project was create a chat bot that uses the discord interface to interact with uses that have a discord account. At the end of this prototype, the chat bot has the following features:

* Stream music from YouTube in to the discord voice channel they a u user is currently in
* Find navigation routes to place users want
* Provide news headlines concerning sports and more from different sources depending on the user's choice
* Give information on sport team such as foundation date, league and the colour of the team's kit

**My Role:**

In this project, apart from helping my team members with some coding problems I was in charge of the following tasks:

* Create the interface between the discord API and the main script of the Chabot
* Bring all together the code written by my team mates to work with together.
* Handle message received from users and determine what action the Chabot would do. An example of this is how I filter commands from common text.
* Used Open Weather API to retrieve weather information according to the user's location.

**How I does it work?:**

Because we wanted the Chabot’s main interface to be discord, we had to import it and create a discord client object as such:

TESS=discord.Client() #--where TESS is the client object that we create.

TESS.run() #--Using the discord run function that connects the machine running the scripts to the discord server placed at the end of the script.

Now that the bot is up and running, I used the *@Client.event* function provided by discord API to detect incoming messages. To be able to differentiate if the text received what a command or general text I wrote the following code:

**if** ((message.content[**0**])=='!' **and** ('play' **in** message.content)): #--checking if an input is a command

**if** TESS.is\_voice\_connected(server) == False: #--checking if we already have a voice connection to the voice client

await TESS.join\_voice\_channel(channel)

TESS\_voice=TESS.voice\_client\_in(server)

**print**('joined voice channel')

message.content=message.content.strip('!')

message.content=message.content.strip('play'

first\_link=get\_vid\_link(message.content)

song\_lenght=int(lenght\_song(first\_link))

vid='https://youtube.com{}'.format(first\_link)

song\_player=await TESS\_voice.create\_ytdl\_player(vid) #--Creates a stream player in a new thread in the background from YouTube.

players\_instances[server.id]=song\_player #-- Storing an instance of the song play so that it can accessed later

song\_player.start()

**print**(players\_instances)

**else**:

TESS\_voice=TESS.voice\_client\_in(server)

message.content=message.content.strip('!')

message.content=message.content.strip('play')

message.content=format\_message(message.content)

first\_link=get\_vid\_link(message.content)

song\_lenght=int(lenght\_song(first\_link))

vid='https://youtube.com{}'.format(first\_link)

song\_player=await TESS\_voice.create\_ytdl\_player(vid)

players\_instances[server.id]=song\_player

song\_player.start()

In the above example, if the first character of the in-coming message is '!' and if it contains the string 'play', then I consider this message to be a command. In this case, the command is to play a song. I used the same method to identify other commands such as stop and resume.

Now that I know the message is a "play” command, I checked if the there was a voice connection between the user and the bot in other to stream audio to the channel the user is in. I removed the command prefix ("!" and "play") from the message and passed it to the format\_message function which remove characters such as question marks. in order to play the correct song required by the user, I wrote a get\_vid\_link function that uses requests and BeautifulSoup modules to search for a song on YouTube and returns the id of the first e video that appears on a YouTube search:

**def** **get\_vid\_link**(artist): #--the argument passed to this function could also be a song.

'''this function that takes in a string and searches it on youtube's website. It gets the first link on the page and return a video id '''

url="https://www.youtube.com/results?search\_query=" + artist

response=requests.get(url)

links=response.text #--converting response object to text:

data=BeautifulSoup(links,'html.parser')#--convert text in html format

list\_links=[]

**for** tag **in** data.find\_all('a'): #--retrieving every link from the <a> tag in data

hf=tag.get('href')

**if** '/watch?v' **in** hf:

list\_links.append(hf)

**if** len(list\_links)==**1**:

**return**(list\_links[**0**]) #--selecting the first link.

**break**

**else**:

**return** False

***Why does it return only the first link?*** Well based on youtube's searching algorithm which is very accurate, the first link on the page would very likely be the song or artist’s most popular song (this will be the case if the user only enters the name of an artist.) the user requested. This can of course be improved in later prototypes by giving the choice to the user to select between the first five results returned by the YouTube search engine.

**What I have learnt while working on this project?**

Throughout the course of this project I have gained many valuable skills. The first one is to be able to work within a team. I always thought it was easier to take on projects individually but doing this group project made me realise that in reality the best work is often done as a team. This is because as a team we learn from each other and complement each other’s skills. Secondly, having to work with a deadline improved my time management skills. This was very crucial because I needed to prioritize time spent on some features of the Chabot and implement them within a fixed amount of time. It also helped me to put into practice what I have learnt about agile development. In addition to that, being a team leader in this project helped me development leadership skills. This was the case because I had to make sure everyone knew what they had to do, if we were on Woking on schedule or not.

Apart from personal development, I also improve on my technical skills. For example, I in order to get data about the weather I had to query an API and filter the data I needed. I also had a brief introduction to Asynchronous programming in python and how powerful it is in designing and optimizing programs.

|  |
| --- |
| **Python Code** |

<https://github.coventry.ac.uk/demanouh/CHATBOT>

<Replace this text with all the code you personally contributed to the project>

(Never use screenshots of your code. Your code should be presented as text. There are many good tools to help you format your code such as <http://hilite.me>)