Félix Lamarche  
Technologies émergentes  
420-5A3-VI gr. 00001

Rapport de projet de session  
Technologies émergentes projet bot Discord.

Travail présenté à  
Étienne Rivard

Département des Techniques de l’informatique  
Cégep de Victoriaville  
Date de remise (Le 31 novembre 2023)

Technologies émergentes projet bot Discord.

<https://github.com/2137772-Lamarche-Felix/Projet-Discord>

## Présentation des buts du projet :

Le but de ce projet était de construire à partir d’aucune connaissance un bot Discord permettant d’associer des rôles et créer des salons textuels en liens avec ceux-ci. J’ai voulu faire un bot un peu comme le bot populaire MEE6[[1]](#footnote-1) bien connu sur discord. Bien sûr, MEE6 est plus complet que mon projet que j’ai fait en 15 semaines, mais j’ai essayé de garder les fonctions de base et de les faire à ma sauce.

## Liste des concepts :

Bot :

En premier, j’ai cherché pour avoir une bonne idée de ce qu’est un bot. J’ai compris que c’est : « Un bot informatique est un agent logiciel automatique ou semi-automatique qui interagit avec des serveurs informatiques. »[[2]](#footnote-2)

Bot Discord :

Après, j’ai cherché plus en profondeur les bots Discord les plus populaires pour m’inspirer pour mon projet. C’est à ce moment que j’ai trouvé MEE6 que j’ai parlé plus haut.

Python :

Il était maintenant temps de trouvé un langage pour coder ce projet de bot discord. J’ai alors choisi python après avoir vu qu’il existait un module sur python utilisant l’API de discord. Discord.py est le nom du module qui fait appel à l’API de discord.

Discord Api :

La plateforme de développement Discord vous permet de créer des applications pour personnaliser et étendre Discord pour des millions d'utilisateurs. Un ensemble d'API et de ressources s'associent parfaitement à une communauté de soutien pour vous aider à construire des robots, intégrer des services tiers et ajouter le support Discord à votre jeu.

## Références

1-<https://discord.com/developers/docs/intro>. C’est le site web de discord qui permet de crée ton bot de base. Sans ça, il serait impossible de pouvoir associer le bot avec notre serveur discord.

2-<https://builtin.com/software-engineering-perspectives/discord-bot-python> Ce site est le tutoriel de base que j’ai utilisé pour la partie création initiale du bot et du projet. Ce site m’a permis de bien comprendre la base de la création de bot, mais surtout de la base de Discord.py qui peut être dur à suivre par bout vu les possibilités presque qu’infini.

3-<https://discordpy.readthedocs.io/en/stable/api.html>. Ce site est la documentation officielle pour Discord.py. C’est le module qu’utilise mon projet alors j’ai eu beaucoup recourt à la documentation de pour mieux comprendre celui-ci.

4-<https://emojipedia.org>. Cette référence est plutôt étrange vu mon projet je sais bien, mais elle m’a été bien utile. J’ai du l’utilisé lorsque je codais la partie pour la gestion de rôle. En résumé, mon bot peut ajouter des rôles grâce à la réaction de l’utilisateur et pour les réactions j’ai dû mettre des emojis. Lorsque j’essayais d’écrire le nom de l’emoji en string ça ne marchait pas, alors j’ai dû copier-coller l’émoji directement dans me code ce qui n’est pas habituel.

5-<https://stackoverflow.com>. StackOverflow est indispensable pour n’importe quel programmeur qui veut en apprendre plus sur une nouvelle technologie. Ce site m’a sauvé la vie plusieurs fois lorsque la documentation officielle était plus ou moins claire ou lorsque j’avais une erreur avec un message d’erreur que je n’avais jamais vu.

## Cartographie

Une image contenant texte, diagramme, reçu, ligne

Description générée automatiquement

## Tableau comparatif avec d'autres technologies

Comme on peut voir sur ce site, <https://discordapi.com/unofficial/comparison.html> il a beaucoup de langage possible pour faire un bot discord. Parmi tous ces choix de langages, 3 sont ressortie : discord-rs en Rust, Discord.net en C# et discord.py.

Un des langages qui me tentait le plus était le Rust. C’est un langage que nous avons exploré dans aucun cours lors de notre technique. J’ai alors pensé que ça serait parfait pour mon projet, mais le problème c’est que discord-rs qui est en Rust ne permet pas de créer, modifier et supprimer des rôles d’utilisateurs. J’ai donc enlevé ce langage de mes choix, car je veux pouvoir faire la gestion des rôles grâce aux commandes.

Mon deuxième choix était Discord.Net qui est en C#. J’ai regardé un peu documentation et ça avait l’air plutôt simple d’utilisation, il a même un tutoriel bien détaillé directement dans la doc[[3]](#footnote-3). Le seule petit truc qui m’a fait changer d’idée c’est que le cours sert à découvrir une nouvelle technologie et cela fait plus de 2 ans que nous faisons du C# à chaque session, alors j’ai préféré changer de langage pour encore plus apprendre.

Je suis donc allé pour mon dernier choix, Python. Nous avons travaillé sur python pendant quelque heure en Nano-ordinateur et en Support-Technique, mais je voulais vraiment approfondir mes connaissances sur ce langage. Python est un langage dans lequel on peut faire n’importe quoi, alors c’est toujours cool de se pratiquer sur ce langage, cela va me servir dans le futur. Python reste un langage plutôt simpliste sur lequel on peut faire des choses poussés. Je ne savais pas trop par où commencer lorsque j’ai choisi je suis donc allé sur la page de documentation de Discord.py et il avait une introduction bien simple super bien expliqué. De plus, sur Python c’est super simple de télécharger un module et l’utilisé avec l’import. Après avoir effectué plusieurs recherches sur la création de bot j’ai aussi vu que Discord.py avec Python était parmi les plus utilisés, donc je me suis dit que si jamais j’ai un problème plus spécifique ça serait plus simple de répondre à ma question avec internet.

## Journal des enjeux et solutions

Premier problème rencontré : Ce probleme était dans les débuts, j’essayais de coder que le bot écrive dans le salon textuel de bienvenue lorsque quelqu’un rejoignait le serveur Discord. Mon probleme était que le bot écrivait toujours dans les messages privés de la personne et non dans le salon textuel de bienvenue. Après plusieurs consultations sur le net, j’ai vu que j’avais oublié de d’ajouter ce code au début ce qui faisait que le bot n’avait pas les droits d’écrire dans le serveur discord :

intents = discord.Intents.default()

intents.members = True # Enable the GUILD\_MEMBERS intent

intents.message\_content = True

# Crée une instance de la classe Client qui représente le bot

bot = commands.Bot(command\_prefix='!', intents=intents)

Solution trouvé sur : <https://stackoverflow.com/questions/73362307/discord-py-on-member-join-sending-a-message-into-a-specific-channel>

Deuxième problème rencontré : Pour mon bot, je voulais qu’il soit possible de s’ajouter les droits d’un rôle grâce à la reaction sur un message. Exemple, mettre une reaction avec un fusil pour avoir le rôle de Counter Strike qui donne accès à un salon textuel seulement pour les gens avec ce même rôle. C’était relativement simple comprendre comment instaurer la fonctionnalité et la mettre en action, mais j’ai eu un problème en arrivant pour mettre quel emoji était lié à quel rôle. Pour faire simple sur discord il a une banque d’émoji disponible de base alors c’est eux que je voulais utiliser, de la façon que les emojis sont utilisé c’est que tu dois mettre deux points avant et après le nom de l’emoji. Exemple, :gun: pour le pistolet à l’eau. Je pensais qu’il était possible de juste mettre cette valeur dans le string du salon.add\_reaction(), mais cela ne marchait pas pour une raison que j’ignore. Alors j’ai effectué quelque recherche et j’ai trouvé qu’il était possible de mettre l’emoji directement dans le string en copiant collant celui-ci du site[[4]](#footnote-4). Après quelque test j’ai vu que j’avais trouvé la solution, car ça marchait maintenant comme je le voulais. Seul petit point négatif, c’est que ce n’est pas super beau de voir des emojis dans du code, alors si j’avais eu plus de temps j’aurait trouvé une autre solution.

Troisième problème rencontré : Pour que les roles ne servent pas a rien j’ai décidé de créer un salon textuel par rôle pour que seulement les gens ayant ce rôle puisse le voir et parler dans celui-ci. Mon problème était que même si je donnais les droits au rôle que je voulais qui aille accès, les autres ayant pas le rôle le voyait quand même. Alors ma solution a été de créer un rôle appelé peuple pour tout le monde qui rejoin. Donc dans mon on\_member\_join, j’ajoute le membre qui vient de rejoindre directement dans le role peuple : await member.add\_roles(discord.utils.get(guild.roles, name="Peuple"). Maintenant qu’il est créé je dois enlever la visibilité pour ce role sur les salons textuels des autres roles. Pour se faire, j’ai dû enlever la possibilité de lire et écrire des messages dans les salons pour le role peuple ce qui fait que tous les gens ayant ce roles ne voit plus les salons des autres roles. Exemple pour le salon de League Of Legend,

channelLol = await guild.create\_text\_channel('league-of-legend') Je créer le salon

await channelLol.set\_permissions(rolePeuple, read\_messages=False,send\_messages=False) J’enlève les permitions pour le peuple

await channelLol.set\_permissions(roleLol, read\_messages=True,send\_messages=True) J’ajoute les permissions pour le rôle League Of Legend

## Conclusion

Pour conclure, j’ai réussi à accomplir mon but de créer un bot discord permettant d’ajouter et de retirer des roles dans un serveur Discord. De plus, mon deuxième but était d’approfondir mes connaissances en python ce qui est réussi pour le temps que j’ai passé sur mon projet je commence à être très à l’aise avec celui. J’avoue que le fait que l’indentation est super importante me faisait peur un peu parce que j’ai souvent de la misère avec celle-ci, mais un fois qu’on est habitué ça se passe super bien. Je suis content d’avoir réussi à faire un projet fonctionnel pour le cours. J’avais peur de rien avoir, mais finalement mon projet c’est bien passé malgré les quelques problèmes que j’ai eus. Mon projet n’est pas super gros, mais je suis content d’avoir livré ce que je voulais faire et surtout le faire en comprenant tout le code que j’ai fait. Je pense que les bots Discord ont un bon potentiel dans le futur. Par exemple, MidJourney l’intelligence artificielle qui permet de générer des images à partir de texte qu’on lui envoie est disponible seulement sur discord. Il est accessible sur discord sur leur serveur public. Ils ont aussi créé un bot permettant d’avoir accès à Midjourney sur n’importe qu’elle autre serveur discord. Cela est une preuve que certaine compagnie commence à de plus en plus utiliser Discord comme plateforme et faire des bots utilisables partout.

1. https://mee6.xyz/en [↑](#footnote-ref-1)
2. https://fr.wikipedia.org/wiki/Bot\_informatique [↑](#footnote-ref-2)
3. https://discordnet.dev/guides/introduction/intro.html [↑](#footnote-ref-3)
4. <https://emojipedia.org> [↑](#footnote-ref-4)