



C'est une communication simple entre un client et un serveur. Les informations sont envoyées sur le serveur depuis le client, et le serveur gère tout ce qui est attente de connexion avec un client. C'est le **serveur qui gère les commandes reçus** car ils doivent de toute façon être exécuté sur la machine serveur.

Pour ce qui est du client nous avons ici une interface graphique simple. Il est possible de prendre un fichier avec une liste (csv ou texte les deux m'on l'aire de fonctionner) d'adresses des serveurs, il faut ensuite remplir le champ de l'IP avec les adresses proposé au-dessus, et le port (il est normalement déjà remplis) il suffit de se connecter.

Nous pouvons déjà choisir une des commandes près remplis Les commandes « kill », « disconnect » et « reset » fonctionne partiellement. Si les commandes citées précédemment sont utilisé au début elles fonctionnent, mais s'il y a eu d'autres commandes il est fortement probable que ça ne passe pas, ce qui revient à fermer l'application avec les conséquences d'un crash du serveur.



Pour l'architecture du code, il y a logiquement l'attente de connexion du serveur, puis la demande de connexion du client au début de chaque code. À la suite de la connexion, du coté serveur nous arrivons déjà à la partie où il y a le traitement des commandes dans le serveur, elles sont exécutées dans des « if » et « elif » pour enchaîner les commandes à la suite qui sont exécuté à l'aide de « psutil » et « subprocess », le tout dans une boucle « tant que » car s'il y a le message « kill », « disconnect » et « reset » le serveur est sensé quitté la boucle est coupée la connexion.

Du côté du client, suite à la connexion dite précédemment nous avons des « def » qui permet d'envoyer et de réceptionner les messages. Le tout dans une seule classe.

Puis nous avons la deuxième classe qui nous permet de faire l'interface graphique. Ici nous pouvons configurer à l'aide de PyQt5 des boutons, liste, champs de texte... et placer en forme de grillage sur une application. Nous pouvons également toucher au style de l'application en injectant un code CSS car c'est pris en charge.

C'est également ici aussi qu'il y a beaucoup de paramètres qui sont réglables avec les actions que les boutons/envoi de messages peuvent faire, ainsi que relier la connexion au GUI.

Pour les tâches fournies j'ai seulement fait la première partie du cahier des charges, avec la connexion du client aux serveurs, lecture d'un fichier pour les adresses IP, l'envoi des commandes.

Je suis conscient que ce n'est pas un travail très fourni, j'ai mis beaucoup de temps à m'impliquer sérieusement au projet, au premier abord je ne l'ai pas trouvé intéressant, je pense que mon niveau en programmation python en général doit beaucoup jouer là-dedans. Beaucoup de temps perdu sur des erreurs car je n'arrivais pas à localiser le problème dans le code des problèmes qui ne sont toujours pas résolus pour certains d'ailleurs.