

Rapport du projet ChatOS

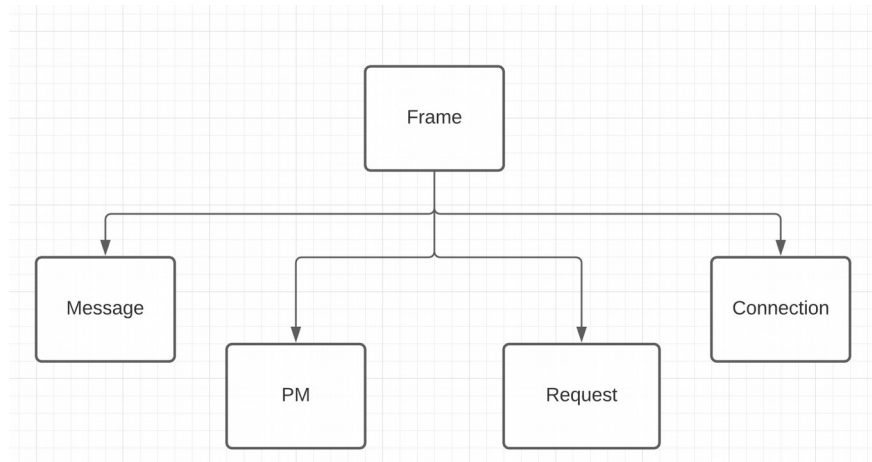
Choix d'architecture :

Pour ce projet, j'ai tout d'abord tenté de travailler en deux étapes.

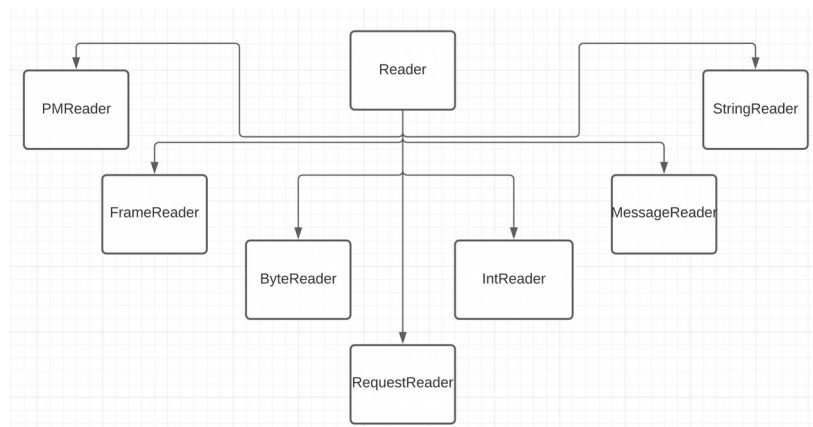
La première consistait à faire le maximum avec une architecture relativement grotesque qui n'est pas très malléable.

Ensuite, une fois que tout aurait fonctionné, j'aurais voulu refaire l'entièreté de la structure du projet pour me rapprocher de quelque chose de plus facilement utilisable, en utilisant des visiteurs, etc.

La structure des frames se fait comme suit :



La structure des readers se fait comme suit :



Ce sont des choix très classiques avec une architecture d'héritage très basique pour les structures de données utilisées.

Par la suite, le serveur et le client fonctionnent de la façon suivante :

Le serveur est en écoute constante des clients pour observer des événements tels que la connexion d'un client ou bien encore la réception d'un paquet envoyé par un client.

Le client a un thread écoutant en permanence l'entrée stdin pour ne rater aucun message envoyé par le client.

En fonction du début du message (s'il commence par un @ ou un /), il effectue l'action associée (message privé ou connexion privé).

Si le message ne commence pas par cela alors le message est considéré comme un message normal.

Le serveur est capable, en même temps, de pouvoir diriger des paquets ou rediriger des paquets qu'il aura lui même écrit.

Si le serveur reçoit un paquet de connexion, il enregistre le client si son pseudo n'est pas encore dans la structure stockant les pseudos de chaque client et renvoie un paquet d'accusation de réception.

Si le client est déjà enregistré, le serveur renvoie un paquet d'erreur et alors le client renverra un paquet.

Les paquets de messages classiques sont transmis à tous les clients connectés au serveur.

Les paquets de messages privés sont envoyés d'un client à l'autre avec une vérification de la redirection nécessaire par le serveur.

Parties fonctionnelles et non-fonctionnelles du projet :

Les messages et messages privés sont fonctionnels.

L'implémentation est très basique et n'est pas la bonne.

J'aurais dû modifier la structure pour obtenir un Reader qui pourrait lire le bytecode de la trame passante.

Les connexions privées ne sont pas fonctionnelles.

De même, mon implémentation était trop brute pour obtenir quelque chose de satisfaisant et non grotesque comme ce que j'ai pu produire.

Les fichiers de tentatives sont cependant disponible dans les sources.

Difficultés rencontrées et évolutions demandées :

Je me suis particulièrement mal organisé pour réaliser ce projet et me suis donc perdu dans la production de mon code jusqu'à obtenir un code fonctionnel avec les concepts de base mais très difficilement modifiable car trop imposant et mal structuré.

Les principales difficultés rencontrées n'ont pas été tant les concepts d'implémentation que je pense avoir compris mais la restructuration du projet qui devait être à la base de la continuation du projet.