Compte rendu projet FAR

Ayoub MOUJANE - Pierre PERRIN - Julien Wiegandt

SPRINT 1 :

Protocole de communication :

Pour ce projet nous avons utilisé les sockets pour qu'un client puisse communiquer avec un autre client en passant par un serveur. Nous avons mis en place une communication en mode connecté (TCP).
Lorsqu'on lance le serveur, celui-ci attend la connexion de deux clients. Une fois connectés, une conversation démarre entre les clients qui s'envoient des messages à tour de rôle. Un message envoyé par un client passe d'abord par le serveur qui le transmet ensuite au client destinataire. La déconnexion se fait lorsqu'un des clients envoie « fin ». Lorsque cela arrive les deux sont déconnectés et le serveur attend la connexion de deux clients pour pouvoir relancer une conversation.
Voir Annexe (page 3) pour le diagramme de séquence.

Difficultés rencontrées :

Un problème rencontré a été le fait que lorsqu'un client envoyait un message il était directement capable d'envoyer un autre message, le problème était que le recv n'était pas bloquant du coup après avoir envoyé un message on n'attendait pas la réception du prochain message. Par conséquent chaque client pouvait envoyer en même temps des messages et plus rien n'allait. Pour régler ce problème on a trouvé sur internet une personne qui proposait comme solution de tester le retour de recv pour savoir si le client avait reçu un message du coup s'il pouvait écrire ou s'il devait attendre.

Lien de la page qui nous a aidé :
https://stackoverflow.com/questions/41077820/c-language-sockets-a-chat-between-two-clients-using-one-server-as-middle-ma

Répartition du travail :

	Initialisation Client	Communication Client	Initialisation Serveur	Communication Serveur	Diagramme séquence
Ayoub Moujane					
Pierre Perrin					
Julien Wiegandt					

Compiler et exécuter le code :

Pour compiler :

Serveur : « *gcc -o serveur serveur.c* »
Client : « *gcc -o client client.c* »

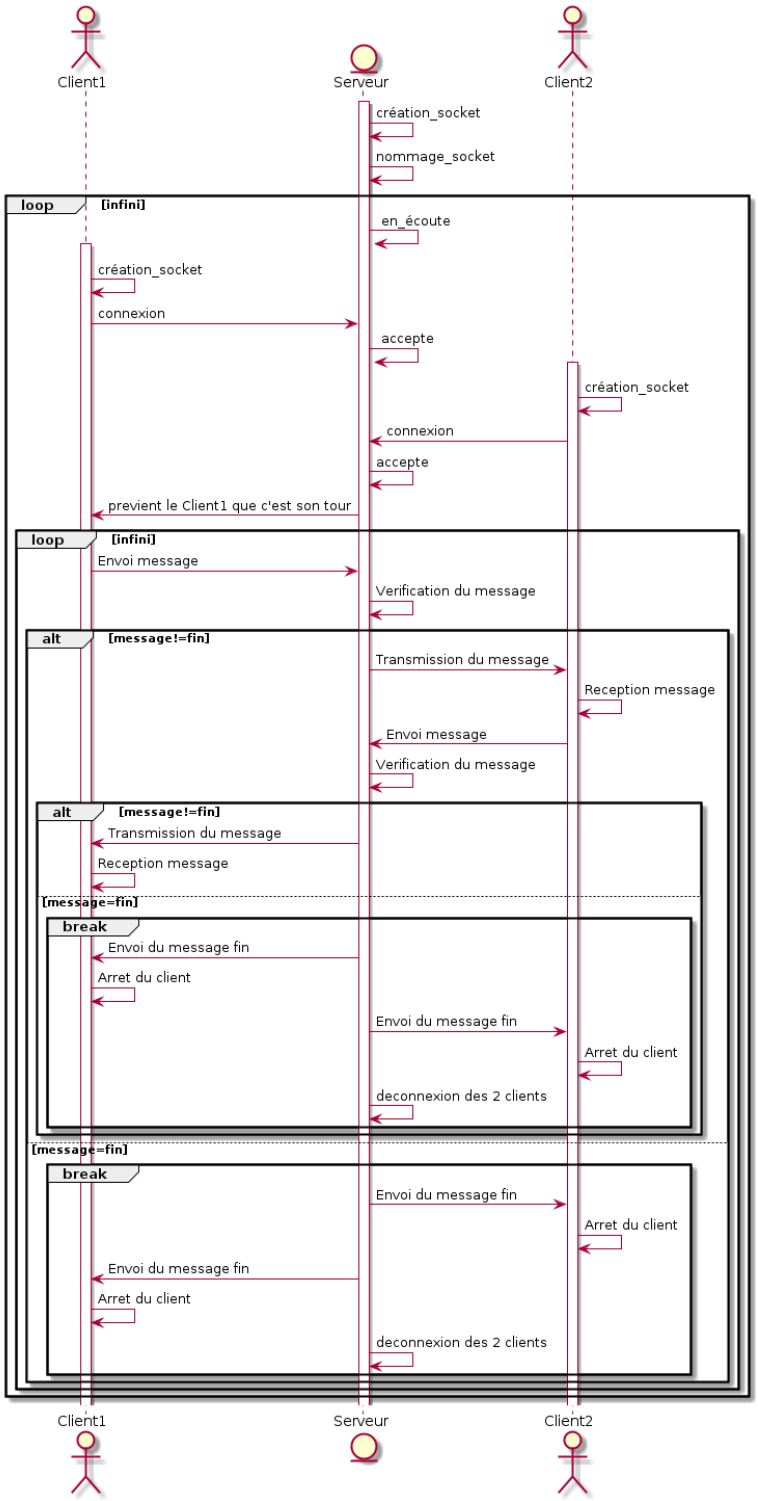
Pour exécuter :

Il faut d'abord lancer le serveur, ensuite le client.
Serveur : « *./serveur* »
Client : « *./client* »

On a choisi d'inclure directement dans le code l'adresse IP ainsi que le numéro de port pour faciliter l'exécution. L'adresse IP est 127.0.0.1 et le numéro de port est 8081.

Annexe :

Diagramme de séquence uml :



Note : En cas d'erreur lors de la création de la socket ou du nommage de la socket par le serveur, celui-ci s'arrête. Nous n'avons pas mis cette information directement dans le diagramme pour éviter de l'alourdir.