TP RMI

Le rendu est constitué de 4 dossiers, 2 dossiers pour la partie polling et 2 dossiers pour la partie callback. ClientRMI et ServeurRMI peuvent permettre de lancer un tchat avec du polling. ClientRMI_Callback et ServeurRMI_Callback servent à la partie Callback.

Les deux tchats fonctionnent et disposent des mêmes commandes. Il suffit de taper « quit! » pour quitter le tchat et rentrer un identifiant quand on se connecte au tchat que les autres clients verront.

RMI Polling :

Le client envoit les messages et le serveur les reçoit et les stocks dans une list, Le clientThread fait du polling et interroge tout le temps le serveur pour savoir si la liste augmente, si oui il récupère le nouveau message et vérifie si ce n'est pas un message qui a été envoyé par la même personne voulant le récupérer, si c'est le cas il ne récupère pas le message.

On voit dans le ClientThread une boucle while faisant du polling, dès qu'il y a un nouveau message il est affiché, mais seulement si la partie identifiant au début du message ne contient pas l'identifiant du client. Par exemple cela permet à l'envoyeur d'un message de récupérer le message qu'il vient d'envoyer.

Le substring permet aussi de simplement comparer le début du « message » contenant l'identifiant, cela évite les problèmes où l'identifiant aurait été dans le contenu du message après la partie identifiant.

Par exemple s'il y a deux clients connectés, Robin et Julien, si Robin envoit un message cela affiche chez Julien : « Robin : Salut ! », mais sans le substring si Robin appelle Julien ce dernier n'aurait pas pu voir le message car cela aurait donné « Robin : Salut Julien ! », donc le message contient l'identifiant de Julien.

Si le client écrit « quit! », il est déconnecté du tchat et cela affiche pour tout les autres clients que l'utilisateur s'est déconnecté. A l'inverse, quand un client se connecte il est aussi affiché pour tout les autres clients qu'un nouveau client est connecté et son identifiant.

RMI Callback :

La partie Callback fonctionne aussi, inspirée du premier projet, elle possède aussi une manière d'identifier les clients et d'envoyer les messages à tous les clients sauf à celui qui a envoyé le message.

Le client écrit un message, le serveur récupère ce message et notifie tous les autres clients du nouveau message, il leur envoi le message avec l'identifiant de l'envoyeur.

Il a donc fallu rajouter le getIdentifiant() dans le ClientListener pour que le serveur ait accès à cette méthode.

Il est aussi possible d'utiliser « quit! » pour quitter le tchat.

Il n'y a pas de possibilité d'avoir accès aux messages envoyés avant la connexion du client.

Les codes ont été commentés pour faciliter la compréhension.