



INF3405 – Réseaux informatiques

Hiver 2020

TP No. 1

Groupe 1

[1928777] – Mariam Sarwat

[1935516] – Louis-Maxime Bois

Soumis à : Bilal Itani

14/02/20

Lors de ce TP, nous avons à mettre en place une communication entre un serveur et un client avec le langage *Java*. Le client a ensuite l'opportunité de passer un fichier photo de type (.jpg, .png ou .gif dans notre cas) au serveur où un filtre, fourni par le chargé de laboratoire, y sera appliqué. Ce fichier filtré sera ensuite retourné au client, renommé à la guise du client et enregistré dans le dossier du client. Il est ensuite possible de se déconnecter de son compte pour se reconnecter ensuite, quitter entièrement l'application ou fournir un autre fichier à filtrer. Le serveur garde en mémoire les informations des utilisateurs et s'assure de l'identité des utilisateurs avec celui-ci. Le but principal de ce TP était de familiariser les étudiants avec les moyens de transfert de fichiers entre deux dossiers et ce, par le biais de *Sockets* et en utilisant des objets intermédiaires comme *BufferedImage*.

Notre travail se résume en trois fichiers distincts, soit le client dans son propre dossier, puis le serveur et l'intermédiaire-client dans un autre dossier. Le client a pour but de faire le contact direct avec l'utilisation et relayer les entrées de l'utilisateur au serveur pour commencer (uniquement pour la mise en place du Socket), mais principalement avec le *ClientHandler*. Il est à noter que nous avons opter pour une approche plus lourde au niveau de client, puisque c'est là que nous effectuons la grande majorité de nos vérifications d'entrées. La logique y est donc assez lourde par rapport au côté serveur.

```
1  import java.net.Inet4Address;
2  import java.net.InetAddress;
3  import java.net.InetSocketAddress;
4  import java.net.ServerSocket;
5  import java.net.UnknownHostException;
6  import java.io.*;
```

Dans le fichier *Server*, les bibliothèques ci-haut ont été appelées. Les bibliothèques 1 et 5 ont servi à implémenter la technique de vérification de la validité des adresses IP venant de *TechieDelight* (<https://www.techiedelight.com/validate-ip-address-java/>) comme cité dans le code. Les bibliothèques 2, 3 et 4 ont servi à la mise en place du socket.

```
1  import java.io.*;
2  import java.net.Socket;
3  import java.util.Scanner;
4  import javax.imageio.ImageIO;
5  import java.awt.image.BufferedImage;
```

Dans le fichier *ClientHandler*, la bibliothèque 3 a été utilisé afin de modifier et parcourir le fichier texte utilisé pour garder en mémoire les informations des utilisateurs. Les bibliothèques 4 et 5 ont servi au passage de tableaux de bytes au format demandé par *Sobel.process()*, puis vers un type de fichier d'image. *ImageIO* est principalement utilisé pour le passage de *bufferedImage* à un type lisible. Les bibliothèques utilisées dans le fichier *Client* ont déjà été abordées.

Nous avons opté après plusieurs essais différents pour effectuer le transfert de fichier entre le client et le serveur à l'aide d'une transition du fichier à envoyer vers un tableau de bytes qui sera transmis position par position de part et d'autre du système. Lors de l'arrivée dans le serveur, l'image ne sera pas complètement reconstituée en fichier lisible avant d'être soumise au filtre de Sobel. Elle sera directement convertie en *BufferedImage*, puis on lui fournira un fichier temporaire pour stocker l'image à la suite de son filtrage. Le même protocole de transfert est réutilisé dans le sens inverse, donc pour acheminer l'image au client.

La même technique de vérification de l'entrée de l'utilisateur a été utilisée sur chaque entrée. Un *boolean* a été initialisé avant les dites entrées et les méthodes *verify()* renvoient tous un *boolean* qui dictera le rythme que prend le déroulement du programme. Des confirmations sont données au client et au serveur au fur et à mesure des opérations concernant le cheminement de l'utilisateur et du fichier soumis à travers le système.

Difficultés rencontrées :

Nous avons eu besoin d'une longue période d'acclimatation à Java, puisque notre dernier contact avec ce langage datait d'un an. Des rappels concernant le comportement de l'opérateur `==` avec les *String*, ainsi que la nature des objets passés en paramètre d'une fonction ont été salutaires après beaucoup de temps passé à chercher. La plupart des problèmes rencontrés portaient sur des détails uniques à Java.

Critiques et Améliorations :

Franchement, le TP, autant son sujet que son exécution, est très intéressant et particulièrement clair. Je ne vois pas vraiment de points assez négatifs pour les soulever.

En somme, ce TP a été une entrée en matière très efficace concernant l'utilisation de *Sockets* et du transfert de fichiers. Le soutien des chargés a été impeccable et le temps donné pour compléter la tâche était plus que nécessaire, mais nous a toutefois permis de se remettre dans le bain avec Java sur une longue période. Ceci est très apprécié.