



École Polytechnique

de Montréal

Département Génie Informatique et Génie Logiciel

INF3405 – Réseaux Informatiques

TP1 : Projet en réseaux informatiques
Traitement d'images par réseau

1. Informations générales

Session	Automne 2020
Public cible	Étudiants de 1 ^{er} cycle
Date et Lieu de réalisation	À partir du 8 septembre 2020 au Laboratoire L-4708
Taille de l'équipe	3 étudiants
Pondération	10 %
Date de remise du projet	Remise pour tous les groupes le 26 octobre (23h55 au plus tard)
Directives particulières	<p>✓ Tout rapport sera pénalisé de 5 points s'il est soumis par une équipe dont la taille est différente de trois (03) étudiants sans l'approbation préalable du chargé de laboratoire.</p> <p>✓ Soumission par <i>moodle</i> uniquement (http://moodle.polymtl.ca).</p> <p>✓ <u>Soumission d'une archive compressée (.zip / .rar) contenant le rapport (en format PDF) et les projets Eclipse contenant le client et le serveur.</u></p> <p>✓ Toute soumission de l'archive en retard est pénalisée à raison de 0.5 points par heure de retard.</p> <p><u>Important:</u> Les laboratoires se feront à distance. Vous serez attendu à partir du 8 septembre à 17h45 dans la chaîne "Général - Annonces" du serveur Discord de votre groupe.</p>
Chargé de laboratoire	Groupe 01/04 - Bilal Itani (bilal.itani@polymtl.ca) Groupe 02/03 - Mehdi Kadi (mehdi.kadi@polymtl.ca) Groupe 05 - Liliane-Caroline Demers (liliane-caroline.demers@polymtl.ca)
Auteurs :	Mehdi Kadi, Bilal Itani

2. Connaissances requises

- Sockets
- Threads
- Programmation **Java uniquement**

3. Objectifs du laboratoire

L'objectif de ce laboratoire est de familiariser l'étudiant :

- aux échanges Client/serveur en utilisant les sockets;
- au développement « d'applications réseau » en utilisant les threads.

4. Description

Introduction :

Suite à des recherches sur Internet, vous avez trouvé un algorithme¹ permettant d'appliquer un filtre de Sobel² sur une image et vous l'avez appliqué sur votre photo de profil Facebook. Vos amis ont apprécié ce filtre et vous demandent de les aider à l'appliquer sur leurs photos de profil. Par conséquent, vous décidez de les aider en leur fournissant un client capable de se connecter à votre serveur capable d'appliquer le filtre de Sobel sur leurs photos de profil. En effet, ce client sera capable d'envoyer une image au serveur et la recevoir une fois que le filtre lui a été appliqué. Cette photo sera d'ailleurs stockée sur leur disque dur afin que vos amis puissent la téléverser sur Facebook.

Pour l'instant, vous prévoyez qu'une simple **interface console**, car vous voulez assurer uniquement le fonctionnement de programme et qu'une interface graphique n'est pas une de vos priorités.

Requis :

Vous aurez à faire 2 produits, un serveur, et un client pour communiquer avec le serveur.

Au démarrage du serveur, celui-ci demande à l'utilisateur d'entrer les informations suivantes : adresse IP du poste sur lequel s'exécute le serveur, port d'écoute (**un port entre 5000 et 5050 uniquement**). Une vérification doit être faite pour s'assurer que l'utilisateur a bien entré des données cohérentes. Par exemple, le serveur doit détecter que l'adresse 7x7.-202.666.888 et le port -9999 sont incohérents. Il devra aussi s'assurer que l'adresse IP entrée est bien sur quatre octets.

Au lancement du client, celui-ci demande à l'utilisateur d'entrer l'adresse IP du serveur, le port du serveur (**entre 5000 et 5050**), un nom d'utilisateur, et un mot de passe. Le client doit vérifier la validité du format de l'adresse IP et du port entrés par l'utilisateur (de la même façon que le fait le serveur) et afficher une erreur en cas d'invalidité. Ces informations sont utilisées pour tenter de se connecter au serveur. Le serveur vérifie alors la correspondance nom d'utilisateur / mot de passe et accepte la connexion si le nom d'utilisateur et le mot de passe est valide. Sinon, il rejette la connexion du client, et le client affiche « Erreur dans la saisie du mot de passe ». Si le nom d'utilisateur n'existe pas dans la base de données locale du serveur, le serveur va créer automatiquement un compte pour l'utilisateur et y assigne le mot de passe. Si la connexion a été acceptée,

¹ <https://stackoverflow.com/questions/41468661/sobel-edge-detecting-program-in-java>

² https://fr.wikipedia.org/wiki/Filtre_de_Sobel

le serveur connecte l'utilisateur au service de traitement d'image. Le client demande ensuite le nom de l'image qui devra être envoyée au serveur. On peut assumer que cette image se trouve dans le même répertoire que l'exécutable du client. Pour des fins de convivialité, le client demande aussi à l'utilisateur le nom de l'image que ce dernier veut attribuer à l'image traitée.

Quand le serveur reçoit une image à traiter de la part d'un client, il doit afficher en console les informations suivantes : [Nom d'utilisateur - Adresse IP : Port client - Date et Heure (min, sec)] : nom de l'image à traiter, par exemple :

[Maxime - 132.207.29.107:46202 - 2020-01-10@13:02:01] : Image photoDeMaxime.jpg reçue pour traitement.

Le client devra avertir l'utilisateur, par un message en console, lorsque l'image a été envoyé au serveur. Le format de cet avertissement est laissé à votre discrétion. Lors de la réception de l'image traitée, le client devra aussi avertir l'utilisateur de cet événement et devra aussi lui indiquer le chemin vers l'image reçue du serveur.

Notez que le programme client et le programme serveur ne sont (en théorie) pas exécutés sur la même machine et que, bien évidemment, plusieurs clients peuvent tenter de se connecter au serveur en même temps. Cependant, chaque client connaît d'avance l'adresse IP du serveur ainsi que le numéro de port écouté.

Le serveur doit aussi pouvoir stocker la base de données contenant les noms d'utilisateur (dans un fichier dans le format de votre choix (e.g .csv/.txt/.json/.xml ou autre...)) des correspondances nom d'utilisateur et mot de passe.

La figure 1 est l'image envoyée par le client au serveur. Ce dernier appliquera sur l'image le filtre de Sobel. La figure 2 est l'image résultante du traitement fait par le serveur, elle sera stockée sur disque une fois qu'elle a été reçue.



Fig. 1 : Les pavillons Lassonde de l'école polytechnique de Montréal (image originale)³



Fig. 2 : Application du filtre de Sobel sur l'image originale

³ <http://www.voirvert.ca/projets/projet-etude/les-pavillons-lassonde-de-l%E2%80%99ecole-polytechnique#LEED>

5. Requis fonctionnels

Les fonctionnalités attendues sont indiquées comme suit :

✓ Serveur

- Saisie des paramètres du serveur (adresse IP, port d'écoute entre 5000 et 5050)
- Pouvoir connecter des utilisateurs avec leurs mots de passe
- Pouvoir fermer le serveur puis le rouvrir, et avoir tous les profils usagers disponibles (écriture sur disque et pas uniquement sur RAM)
- Recevoir des images à traiter
- Appliquer le filtre de Sobel sur l'image
- Tenir une base de données des usagers et leurs mots de passe
- Afficher en temps réel les demandes à traiter
- Effectuer adéquatement la vérification nom d'utilisateur/ mot de passe
- Créer les comptes automatiquement s'ils n'existent pas
- **Le serveur doit utiliser l'algorithme dans le fichier Sobel.java pour effectuer la transformation sur l'image.**

✓ Client

- Saisir au clavier l'adresse IP du serveur et le port d'écoute
- Vérifier la validité de l'adresse IP saisie (uniquement le format) et le numéro de port (entre 5000 et 5050)
- Se connecter au serveur.
- Saisir au clavier le nom d'une image se trouvant dans le répertoire contenant l'exécutable.
- Envoyer au serveur l'image à traiter
- Recevoir et enregistrer sur disque l'image une fois traitée
- Afficher un avertissement dans la console une fois que l'image a été envoyée pour le traitement
- Afficher un avertissement dans la console lorsque l'image traitée a été reçue ainsi que l'emplacement où elle a été sauvegardée.
- Se déconnecter.

6. Livrable

1. Langages et bibliothèques autorisés

- Le client et le serveur doivent être développés uniquement en Java.
- Usage permis de bibliothèques externes comme JSON.simple ou GSON. Si vous utilisez ces bibliothèques, veuillez justifier dans la section « Présentation » de votre rapport la raison de leur utilisation.

Tout non-respect de ces consignes entraînera la note de 0...

2. Soumission

Le livrable est une archive (ZIP ou RAR) ayant le format suivant:

INF3405_TP1_matriculeX_matriculeY où matriculeX > matriculeY

Votre archive contiendra les fichiers suivants :

- Les projets Eclipse du client et du serveur incluant les fichiers sources (.java), autrement dit le dossier en entier contenant votre projet
- Le rapport au format PDF
- Les fichiers exécutables (.jar)

****Assurez-vous que les livrables compilent et s'exécutent adéquatement sur les ordinateurs du laboratoire ! ****

3. Rapport

Le rapport d'une longueur maximale de 3 pages (excluant la page de présentation) doit comporter les éléments suivants :

- **Page présentation** qui doit contenir le nom ou le logo de l'école, le libellé et l'identifiant du cours, la session, le numéro et l'identification du projet, la date de remise, les matricules et noms des membres de l'équipe, la mention « Soumis à : **nom et prénom du chargé de laboratoire** ».
- **Introduction** en vos propres mots pour mettre en évidence le contexte et les objectifs du TP.
- **Présentation** de vos travaux. Une explication de votre solution mettant en lumière la prise en compte des principaux requis du système. Si vous utilisez des configurations particulières des bibliothèques ou des projets, précisez-les également.
- **Difficultés rencontrées** lors de l'élaboration du TP et les éventuelles solutions apportées.
- **Critiques et Améliorations** : Il serait intéressant d'inclure vos suggestions pour améliorer le laboratoire.

- **Conclusion :** Expliquez en quoi ce laboratoire vous a été utile, ce que vous avez appris, si vos attentes ont été comblées, etc.

7. Évaluation

Évaluation de l'exécutable	6
Évaluation de l'implémentation : gestion adéquate des variables et de toute ressource (création, utilisation, libération), gestion des erreurs, logique de développement, documentation du code , etc.	8
Rapport	6
Total des points	20