

École Polytechnique de Montréal
Département Génie Informatique et Génie Logiciel

INF3405 – Réseaux Informatiques

TP2 : Projet en réseaux informatiques
Traitement d'images par réseau

1. Informations générales

Session	Été 2018
Public cible	Étudiants de 1 ^{er} cycle
Date et lieu de réalisation	À partir du 10 mai 2018 au laboratoire L-4708
Taille de l'équipe	2 étudiants
Pondération	10 %
Date de remise du projet	30 mai 2018 (23h55 au plus tard)
Directives particulières	<ul style="list-style-type: none">✓ 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.✓ Soumission par <i>moodle</i> uniquement (http://moodle.polymtl.ca).✓ Soumission d'une archive compressée ZIP ou RAR contenant le rapport en format PDF et les projets Eclipse contenant le client et le serveur.✓ Toute soumission de l'archive en retard est pénalisée à raison de 0.5 points par heure de retard.
Chargé de laboratoire	Gabriel-Andrew Pollo-Guilbert (gabriel.pollo-guilbert@polymtl.ca)
Auteurs	Mehdi Kadi, Bilal Itani
Révision	Gabriel-Andrew Pollo-Guilbert

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 réseaux » en utilisant les threads.

Ce travail pratique consiste, par la même occasion, à évaluer deux des 12 qualités de l'ingénieur définies par le BCAPG (Bureau canadien d'agrément des programmes de génie). Le Bureau d'agrément a pour mandat d'attester que les futurs ingénieurs ont atteint ces 12 qualités à un niveau acceptable.

Qualité 4 (Conception) : capacité de concevoir des solutions à des problèmes d'ingénierie complexes et évolutifs et de concevoir des systèmes, des composants ou des processus qui répondent aux besoins spécifiés, tout en tenant compte des risques pour la santé et la sécurité publiques, des aspects législatifs et réglementaires, ainsi que des incidences économiques, environnementales, culturelle et sociales.

Qualité 7 (Communication) : habileté à communiquer efficacement des concepts d'ingénierie complexes, au sein de la profession et au public en général, notamment lire, rédiger, parler et écouter, comprendre et rédiger de façon efficace des rapports et de la documentation pour la conception, ainsi qu'énoncer des directives claires et y donner suite.

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.

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

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

Lors de votre premier laboratoire en réseaux informatiques vous avez été introduits aux sockets et aux *threads*, c'est donc l'occasion ultime d'épater vos amis et surtout de leur rendre service.

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 le port -9999 et l'adresse 7x7.-202.666.888 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, 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 selon le format suivant :

```
[Utilisateur - Adresse IP:Port client - Date@Heure] : nom de l'image à traiter
```

Voici un exemple :

```
[Maxime - 132.207.29.107:46202 - 2018-01-10@13:02:01] : photoDeMaxime.jpg reçue
```

Le client devra avertir l'utilisateur, par un message en console, lorsque l'image a été envoyée 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 (CSV, TXT, JSON, XML, etc.) 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.



Figure 1 : Les pavillons Lassonde de l'École Polytechnique de Montréal³



Figure 2 : Application du filtre de Sobel

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

5. Requis fonctionnels

Qualité évaluée

4.3 Procéder à la conception

Critère d'évaluation : Intégrer les concepts de programmation en réseautique retenus au premier laboratoire en répondant aux besoins et en respectant les requis fonctionnels du projet courant.

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
- Tenir une base de données des usagers et leurs mots de passe
- Persistance des profils des usagers entre plusieurs exécutions
- Recevoir des images à traiter
- Appliquer le filtre de Sobel sur l'image
- Afficher en temps réel les demandes à traiter
- Effectuer adéquatement la vérification de l'utilisateur avec son mot de passe
- Créer des comptes automatiquement s'ils n'existent pas

✓ **Client**

- Saisie de l'adresse et du port du serveur
- Vérifier la validité de l'adresse et du port (entre 5000 et 5050)
- Pouvoir se connecter au serveur
- Saisie du nom d'une image se trouvant dans le répertoire de l'exécutable
- Envoyer au serveur l'image à traiter
- Afficher un avertissement dans la console une fois que l'image a été envoyée
- Afficher un avertissement dans la console lorsque l'image traitée a été reçue
- Déconnexion du client

6. Livrable

Langages et les bibliothèques autorisés

- Le client et le serveur doivent être développés uniquement en Java
- L'utilisation des 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.

Qualité évaluée

7.1 Lire et rédiger de la documentation

Critère d'évaluation : Rédiger un rapport technique documentant efficacement le travail d'ingénierie réalisé dans ce projet en utilisant différentes formes de langage (naturel, informatique, etc.).

Rapport

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

- Une **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 ».
- Une **introduction** en vos propres mots pour mettre en évidence le contexte et les objectifs du TP.
- La **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.
- Les **difficultés** rencontrées lors de l'élaboration du TP et les éventuelles solutions apportées.
- Des **critiques et améliorations** de ce laboratoire afin de l'améliorer.
- Une **conclusion** dans laquelle vous expliquez en quoi ce laboratoire vous a été utile, ce que vous avez appris, si vos attentes ont été comblées, etc.

Soumission

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

INF3405_TP2_matriculeX_matriculeY où matriculeX > matriculeY

Votre archive contiendra les fichiers suivants :

- Le projet Eclipse du client et du serveur incluant les fichiers sources Java

- Le rapport en format PDF
- Les fichiers exécutables en format JAR

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

7. Évaluation

Évaluation de l'exécutable	6
Évaluation de l'implémentation (gestion adéquate des variables, des ressources, gestion d'erreurs, logique de développement, documentation du code , etc.)	8
Rapport	6
Total des points	20