Projet de Développement Logiciel

L3 Informatique – Université de Bordeaux

1 Introduction

Ce document présente les besoins nécessaires au développement d'une application de traitement d'image avec une architecture de type client-serveur. Afin de permettre aux groupes d'approfondir un sujet qui les intéresse plus particulièrement, on propose la structure suivante.

Noyau commun : Chaque groupe devra réaliser l'implémentation du fonctionnement central de l'application client-serveur. Elle est décrite dans la section 3 et consiste principalement à articuler le contenu développé durant les premières semaines au sein des TP communs.

Le code de cette partie fera l'objet d'un rendu intermédiaire.

Extensions : Des suggestions d'extensions seront proposées en cours (répartition des charges entre client et serveur, amélioration de l'interface utilisateur, traitements d'image plus avancés, généricité des algorithmes).

Chaque groupe peut faire évoluer ce document avec l'aval de son chargé de TD. Le cahier des besoins fera partie des rendus.

L'application devra permettre de traiter les images en niveau de gris et en couleur enregistrées aux formats suivants :

— `JPEG`
— `PNG`

2 Noyau commun

2.1 Serveur

Besoin 1 : Initialiser un ensemble d'images présentes sur le serveur

Description : Lorsque le serveur est lancé, il doit enregistrer toutes les images présentes à l'intérieur du dossier `images`. Ce dossier `images` doit exister à l'endroit où est lancé le serveur. Le serveur doit analyser l'arborescence à l'intérieur de ce dossier. Seuls les fichiers images correspondants aux formats d'image reconnus doivent être traités.

Gestion d'erreurs : Si le dossier `images` n'existe pas depuis l'endroit où a été lancé le serveur, une erreur explicite doit être levée.

Tests :

1. Lancement de l'exécutable depuis un environnement vide, une erreur doit se déclencher indiquant que le dossier `images` n'est pas présent.

Besoin 2 : Gérer les images présentes sur le serveur

Description : Le serveur gère un ensemble d'images. Il stocke les données brutes de chaque image ainsi que les méta-données nécessaires aux réponses aux requêtes (identifiant, nom de fichier, taille de l'image, format,...). Le serveur peut :

1. accéder à une image via son identifiant,
2. supprimer une image via son identifiant,
3. ajouter une image,
4. construire la liste des images disponibles (composée uniquement des métadonnées).

Besoin 3 : Appliquer un algorithme de traitement d'image

Description : Le serveur contient l'implémentation des algorithmes de traitement d'image proposés à l'utilisateur (voir partie 3.4).

2.2 Communication

Pour l'ensemble des besoins, les codes d'erreurs à renvoyer sont précisés dans le paragraphe "Gestion d'erreurs".

Besoin 4 : Transférer la liste des images existantes

Description : La liste des images présentes sur le serveur doit être envoyée par le serveur lorsqu'il reçoit une requête `GET` à l'adresse `/images`.
Le résultat sera fourni au format `JSON`, sous la forme d'un tableau contenant pour chaque image un objet avec les informations suivantes :

Id : L'identifiant auquel est accessible l'image (type `long`)

Name : Le nom du fichier qui a servi à construire l'image (type `string`)

Type : Le type de l'image (type `jpg,png`)

Size: `Une description de la taille de l'image, par exemple 640*480*3 pour une image en couleur de` 640×480 `pixels (type string)`

Tests : Pour le dossier de tests spécifié dans Besoin 1, la réponse attendue doit être comparée à la réponse reçue lors de l'exécution de la commande.

Besoin 5 : Ajout d'image

Description : L'envoi d'une requête `POST` à l'adresse `/images` au serveur avec des données de type multimedia dans le corps doit ajouter une image à celles stockées sur le serveur (voir Besoin 2).

Gestion d'erreurs :

201 Created : La requête s'est bien exécutée et l'image est à présent sur le serveur.

415 Unsupported Media Type : La requête a été refusée car le serveur ne supporte pas le format reçu (par exemple `EXR`).

Besoin 6 : Récupération d'images

Description : L'envoi d'une requête `GET` à une adresse de la forme `/images/id` doit renvoyer l'image stockée sur le serveur avec l'identifiant `id` (entier positif). En cas de succès, l'image est retournée dans le corps de la réponse.

Gestion d'erreurs :

200 OK : L'image a bien été récupérée.

404 Not Found : Aucune image existante avec l'identifiant `id`.

Besoin 7 : Suppression d'image

Description : L'envoi d'une requête `DELETE` à une adresse de la forme `/images/id` doit effacer l'image stockée avec l'identifiant `id` (entier positif).

Gestion d'erreurs :

200 OK : L'image a bien été effacée.

404 Not Found : Aucune image existante avec l'identifiant `id`.

Besoin 8 : Exécution d'algorithmes par le serveur

Description : L'envoi d'une requête `GET` à une adresse de la forme `/images/id?algorithm=X&p1=Y&p2=Z` doit permettre de récupérer le résultat de l'exécution de l'algorithme `X` avec les paramètres `p1=Y` et `p2=z`. Un exemple plus concret d'URL valide est : `/images/23?algorithm=increaseLuminosity&p1=25`
En cas de succès, le serveur doit renvoyer l'image obtenue après traitement.

Gestion d'erreurs :

200 OK : L'image a bien été traitée.

400 Bad Request : Le traitement demandé n'a pas pu être validé par le serveur pour l'une des raisons suivantes :
- l'algorithme n'existe pas ;
- l'un des paramètres mentionné n'existe pas pour l'algorithme choisi ;
- la valeur du jeu de paramètres est invalide.

Le message d'erreur doit clarifier la source du problème.

404 Not Found : Aucune image existante avec l'indice `id`.

500 Internal Server Error : L'exécution de l'algorithme a échoué pour une raison interne.

2.3 Client

Les actions que peut effectuer l'utilisateur côté client induisent des requêtes envoyées au serveur. En cas d'échec d'une requête, le client doit afficher un message d'erreur explicatif.

Besoin 9 : Parcourir les images disponibles sur le serveur

Description : L'utilisateur peut visualiser les images disponibles sur le serveur. La présentation visuelle prendra la forme d'une galerie d'images. La vignette contenant l'image prendra une taille maximale en fonction de la hauteur et de la largeur de la page. Suivant la taille de l'image initiale la vignette sera complètement remplie en hauteur ou en largeur.

Besoin 10 : Sélectionner une image et lui appliquer un effet

Description : L'utilisateur peut se rendre sur un onglet « Filter », sélectionner une image dans le menu déroulant pour lui appliquer un des filtres proposés de son choix. Il peut être amené à préciser les paramètres nécessaires au traitement choisi (voir partie 3.4). L'image après traitement sera alors affichée sur la page.

Besoin 11 : Enregistrer une image sur disque

Description : L'utilisateur peut sauvegarder dans son système de fichier l'image chargée, avant ou après lui avoir appliqué un traitement.

Besoin 12 : Ajouter une image aux images disponibles sur le serveur

Description : L'utilisateur peut ajouter une image choisie dans son système de fichier aux images disponibles sur le serveur. Cet ajout n'est pas persistant au redémarrage du serveur (il n'y a pas d'ajout de fichiers côté serveur).

Besoin 13 : Suppression d'image

Description : Le client peut choisir de supprimer une image préalablement sélectionnée. Elle n'apparaîtra plus dans les images disponibles sur le serveur. Cette suppression n'est pas persistante au redémarrage du serveur (il n'y a pas de suppression de fichiers côté serveur).

2.4 Traitement d'images

Besoin 14 : Réglage de la luminosité

Description : L'utilisateur peut augmenter ou diminuer la luminosité de l'image sélectionnée.

Besoin 15 : Égalisation d'histogramme

Description : L'utilisateur peut appliquer une égalisation d'histogramme à l'image sélectionnée. L'égalisation sera apliquée au choix sur le canal S ou V de l'image représentée dans l'espace HSV.

Besoin 16 : Filtre coloré

Description : L'utilisateur peut choisir de rehausser la teinte de tous les pixels en rouge, bleu ou vert de façon à obtenir un effet de filtre coloré.

Besoin 17 : Filtres de flou

Description : L'utilisateur peut appliquer un flou à l'image sélectionnée. Il peut définir le filtre appliqué (moyen ou gaussien) et choisir le niveau de flou. La convolution est appliquée sur les trois canaux R, G et B.

Besoin 18 : Filtre de contour

Description : L'utilisateur peut appliquer un détecteur de contour à l'image sélectionnée. Le résultat sera issu d'une convolution par le filtre de Sobel. La convolution sera appliquée sur la version en niveaux de gris de l'image.

2.5 Besoins non-fonctionnels

Besoin 19 : Compatibilité du serveur

Description : La partie serveur de l'application sera écrite en Java (JDK 11) avec les bibliothèques suivantes :

- `org.springframework.boot` : version 2.6.0
- `BoofCV` : version 0.39.1

Son fonctionnement devra être éprouvé sur au moins un des environnement suivants :

- Windows 10
- Ubuntu 20.04
- Debian Buster
- MacOS 11

Besoin 20 : Compatibilité du client

Description : Le client sera écrit en JavaScript et s'appuiera sur la version `3.x` du framework `Vue.js`.
Le client devra être testé sur au moins l'un des navigateurs Webs suivants, la version à utiliser n'étant pas imposée :

- Safari
- Google Chrome
- Firefox

Besoin 21 : Documentation d'installation et de test

Description : La racine du projet devra contenir un fichier `README.md` indiquant au moins les informations suivantes :

- Système(s) d'exploitation sur lesquels votre serveur a été testé, voir Besoin 19.
- Navigateur(s) web sur lesquels votre client a été testé incluant la version de celui-ci, voir Besoin 20.

3 Extension

3.1 Serveur

Besoin 22 : Gestion d'album

Description : Ajout de la gestion d'albums. L'utilisateur peut :

- Créer et attribuer un nom à un album
- Ajouter une image disponible à un album
- Afficher les images presentent dans l'album dans un galerie
- Supprimer un album

Besoin 23 : Superposition des filtres

Description : Il est possible d'ajouter un filtre sur une image sur laquelle on a auparavant déjà appliqué un ou plusieurs autres filtres.

Besoin 24 : Application de filtre : Undo / Redo

Description : Lors de la modification d'une image avec un ou plusieurs filtres le serveur doit stocker chaque opérations effectuées.

3.2 Communication

Besoin 25 : Transférer la liste des images disponibles dans un album

Description : La liste des images présentes dans un dossier doit être envoyée par le serveur lorsqu'il reçoit une requête GET à l'adresse /images ?liste=X. Cette liste doit etre de la même forme que pour le Besoin 20.

Besoin 26 : Gestion des albums

Description : Les requetes possible pour gérer les albums sont :
- Créer un album : /images ?create=NomAlbum ;
- Supprimer un album : /images ?delete=NomAlbum ;
- Renvoyer les images disponibles dans un album : /images ?liste=NomAlbum ;
- Ajouter une image à un album : /images ?add=NomAlbum&id=IdImage.

Besoin 27 : Exécutions des fonctions undo et redo par le serveur

Description :
L'envoi d'une requête GET de la forme images/filter/request doit renvoyer en fonction de la demande undo ou redo l'image précédement modifiées ou celle d'après si les conditions suivantes sont respéctées :
- Ne pas pouvoir faire un undo sur la première image si elle n'est pas modifiée ;
- Ne pas pouvoir faire un redo sur la dernière image modifiée ;
- Supprimer toutes les modifications qui sont après la requête undo, lorsqu'on a une suite de requête undo suivi de modification.

3.3 Client

Besoin 28 : Parcourir les dossiers disponibles sur le serveur

Description : L'utilisateur peut visualiser les albums disponibles sur le serveur. La présentation visuelle prendra la forme de plusieurs icônes alignés.

Besoin 29 : Parcours des images dans un album

Description : L'utilisateur peut visualiser seulement les images présentent dans l'album sélectionné. La présentation prendra la forme d'une galerie d'images.

Besoin 30 : Ajout d'un nouvel album sur le serveur

Description : L'utilisateur peut ajouter un nouvel album à l'aide d'un bouton.

Besoin 31 : Ajout d'une nouvelle dans un album sur le serveur

Description : L'utilisateur peut ajouter une nouvelle image dans un album à l'aide d'un bouton.

Besoin 32 : Bouton undo et redo

Description : L'utilisateur peut à l'aide de boutons revenir sur les modications portées à une image.

3.4 Traitement d'images

Besoin 33 : Négativité

Description : L'utilisateur peut appliquer un effet de négatif sur l'image séléctionnée.

Besoin 34 : Miroir Horizontale

Description : L'utilisateur peut appliquer un effet miroir sur la partie horizontale de l'image séléctionnée.

Besoin 35 : Miroir verticale

Description : L'utilisateur peut appliquer un effet miroir sur la partie verticale de l'image séléctionnée.