

Traitement d'image sous Android

Cahier des charges

Fabien Baldacci, Boris Mansencal, Anne Vialard

1 Introduction

Ce document contient une description du travail à rendre dans le cadre du projet de traitement d'image sous Android en 3ème année de licence.

Le projet est à faire par groupes de 4 étudiants, vous devrez donc découper le projet en tâches afin de vous les partager.

2 Description

Le but de ce projet est de développer une application de traitement d'image sur smartphone avec système Android. Les images peuvent aussi bien être obtenues depuis la galerie du téléphone que directement depuis la caméra.

Le projet est découpé en deux parties, et deux releases du logiciel seront donc à rendre. La première release est commune à tous les groupes. Elle est composée des fonctionnalités de base afin de gérer, afficher, traiter et sauvegarder les images. Ces fonctionnalités sont décrites dans la partie 3.1. Avec cette première release, chaque groupe rendra un rapport présentant le travail effectué y compris l'organisation du code et les tests. Le rapport contiendra aussi la liste des fonctionnalités additionnelles choisies pour la release finale.

3 Besoins fonctionnels

Les besoins fonctionnels sont découpés en deux parties, une liste de besoins fonctionnels obligatoires qui forment le contenu de la première release, et une liste de besoins fonctionnels au choix, qui compléteront la première release afin de former la seconde release (le rendu final).

3.1 Obligatoires

1. **Charger une image** L'application doit permettre d'obtenir une image de plusieurs manières :
 - (a) **depuis la galerie** l'application doit permettre de sélectionner une image parmi celles présentes dans la galerie du téléphone.
 - (b) **depuis la caméra** l'application doit permettre de capturer une image depuis la ou les caméra(s) du téléphone.
2. **Afficher une image** Une fois une image chargée, l'application doit l'afficher sur l'écran du terminal
3. **Zoomer** Lorsqu'une image est affichée dans l'application, il doit être possible de zoomer et dézoomer, en utilisant l'interaction avec deux doigts.
4. **Scroller** Lorsqu'une image est affichée et déborde de l'écran, il doit être possible de déplacer la zone affichée à l'aide d'une interaction avec un doigt.

5. **Appliquer des filtres** L'application doit permettre d'appliquer quelques filtrages usuels sur les images chargées.
 - (a) **Régler la luminosité** Lorsqu'une image est affichée il doit être possible d'en régler la luminosité.
 - (b) **Régler le contraste** Lorsqu'une image est affichée il doit être possible d'en régler le contraste.
 - (c) **Égalisation d'histogramme** L'application doit permettre d'égaliser l'histogramme de l'image affichée.
 - (d) **Filtrage couleur** L'application doit permettre de mettre en oeuvre les traitements couleur vus en TP, à savoir la modification de la teinte ainsi que la sélection d'une teinte à conserver lors du passage en niveaux de gris.
 - (e) **Convolution** L'application doit permettre d'appliquer différents filtres basés sur une convolution. Les filtres moyenneur, Gaussien, Sobel et Laplacien seront implémentés.
6. **Réinitialiser** L'application doit permettre de réinitialiser l'image c'est-à-dire d'annuler les effets appliqués depuis le chargement de l'image.
7. **Sauvegarder une image** L'application doit permettre de sauvegarder une image modifiée.

3.2 Au choix

8. **Simuler un effet dessin au crayon** L'application devra permettre à partir d'une photo de générer un dessin au crayon représentant l'image initiale.
9. **Simuler un effet cartoon** L'application devra restreindre les couleurs d'une image et renforcer les contours afin de lui donner un effet de bande dessinée.
10. **Incruster des objets** L'application devra permettre d'incruster des objets sur des points précis d'un visage (yeux, oreilles, tête, nez, bouche)
11. **Schtroumpfer un visage** L'application devra permettre de transformer un visage en Schtroumpf, à savoir bleuter la peau et ajouter un bonnet blanc.
12. **Restreindre la zone d'application d'un filtre avec le doigt** L'application devra permettre de restreindre l'application d'un filtre uniquement autour d'une zone définie par l'utilisateur avec son doigt.
13. Toute bonne idée d'effet validée par les responsables de TD.

4 Besoins non fonctionnels

4.1 Obligatoire

1. **Gestion de version** Le développement devra utiliser l'outil de gestion de version *git*.
2. **Optimisation du code** L'application devra être fluide. On utilisera notamment *renderscript*. Il faut mesurer les gains obtenus par ces optimisations.

4.2 Au choix

3. **Ergonomie** L'application doit dans tous les cas être pratique à utiliser mais vous pouvez travailler particulièrement à l'amélioration de l'ergonomie de l'interface utilisateur.

5 Structuration du code

Vous devrez réfléchir à une structuration du code permettant de maintenir efficacement votre application, et d'ajouter des fonctionnalités supplémentaires sans duplication de code.

6 Calendrier

Le projet démarre le 17 Janvier et dure 3 mois. Vous devez envoyer le lien de votre projet git aux chargés de TD par e-mail dès le début du projet en précisant les noms des membres du groupe.

Lors de ce projet, vous devez faire deux rendus obligatoires. Pour chacun des rendus, vous devez rappeler le lien de votre projet git aux chargés de TD par e-mail. Le rapport en format pdf peut être mis sur le git et/ou envoyé par e-mail s'il n'est pas trop volumineux.

L'heure maximale de rendu est 23h59.

6.1 1er rendu - 28 février

Le premier rendu comporte une première release du logiciel ainsi que le rapport.

La première release doit comporter tous les besoins fonctionnels de la section 3.1, mais cela ne doit pas vous empêcher de commencer le développement des fonctionnalités supplémentaires.

Le rapport contiendra une description de votre code : découpage en classes, rôle de chaque classe. Il contiendra également la présentation et les tests des fonctionnalités développées, les bugs et problèmes connus. Il finira par les choix de fonctionnalités supplémentaires qui vous semblent réalisables dans le temps restant.

6.2 Rendu final - 15 Avril

Le rendu final contiendra votre logiciel terminé avec toutes les fonctionnalités que vous aurez choisies. Vous devez aussi fournir une série de tests garantissant le bon fonctionnement de votre logiciel, ainsi qu'une évaluation de ses performances en temps et en mémoire.

Votre rapport doit faire un bilan de l'ensemble du travail réalisé :

- description des fonctionnalités disponibles
- description des algorithmes implémentés en détaillant ceux qui n'ont pas été présentés en cours
- description et justification de l'organisation de votre code
- présentation et explication des résultats des tests et des mesures de performances.