

# Traitement d'image sous Android

## Cahier des charges

Emma BESSE, Cauli HONORÉ, Emma VEILLAT

03-03-17

## 1 Présentation du projet

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.

### 1.1 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 et sauvegarder les images. De plus seront intégrés quelques traitements d'image basés sur une transformation d'histogramme ou une convolution.

### 1.2 Programmes déjà existants

De nombreuses applications existent déjà sur le marché (gratuites ou payantes) pour transformer des images (par exemple Photoshop) ou pour appliquer des filtres à des photos (par exemple Snapchat). Elles ont toutes les mêmes fonctionnalités basiques : mettre en niveau de gris une image, modifier la luminosité ... et ont toutes pour finalité de permettre à leurs utilisateurs de modifier les photos à leur convenance.

### 1.3 Objectifs des développeurs

Ainsi, l'application ne vise pas à révolutionner l'univers du traitement d'images : il s'agit, pour les développeurs, d'un entraînement à la programmation objet et à la programmation XML. Il faudra également savoir réemployer du code déjà existant sur Internet afin de le rendre fonctionnel dans l'application.

### 1.4 À qui s'adresse l'application ?

Le logiciel s'adresse à tout utilisateur souhaitant faire des transformations basiques d'images axées sur les couleurs ou des modifications de leur taille, orientation ...

## 2 Besoins fonctionnels

Les besoins fonctionnels sont donc découpés en deux parties : une liste de besoin 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.

### 2.1 Obligatoires

1. **Charger une image** Le logiciel doit permettre d'obtenir une image de plusieurs manières :
  - (a) **depuis la galerie** Le logiciel doit permettre de sélectionner une image parmi celles présentes dans la galerie du téléphone. Pour ce faire, une activité permettant d'accéder à la galerie est créée. Ainsi, l'utilisateur choisit l'image qu'il souhaite modifier et l'application peut la charger grâce à son chemin d'accès.
  - (b) **depuis la caméra** Le logiciel doit permettre de capturer une image depuis la ou les caméra(s) du téléphone. Une activité permettant l'accès à la caméra est créée. Une fois la photo prise et approuvée par l'utilisateur, celle-ci s'affiche dans la galerie, où elle est ainsi enregistrée parmi les autres images déjà présentes.
2. **Afficher une image** Une fois une image chargée, le logiciel doit l'afficher sur l'écran du terminal.
3. **Zoomer** Lorsqu'une image est affichée dans le logiciel, il doit être possible de zoomer et dézoomer, en utilisant l'interaction avec deux doigts. Pour la première release, seul un menu permettra à l'utilisateur d'observer l'image qu'il a choisi avec le zoom (pinch zoom). Par la suite, l'utilisateur pourra utiliser cette fonctionnalité dans toute l'application.
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** Le logiciel 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. Un des menus permet donc de modifier la luminosité d'une image (et d'autres caractéristiques) à l'aide de réglettes. La transformation de l'image est visible en temps durant la manipulation de ces réglettes.
  - (b) **Régler le contraste** Lorsqu'une image est affichée il doit être possible d'en régler le contraste. Un menu permet de faire une extension de dynamiques, ce qui aura pour effet d'étirer l'histogramme de l'image affichée, et ainsi de régler le contraste de l'image.
  - (c) **Égalisation d'histogramme** Le logiciel doit permettre d'égaliser l'histogramme de l'image affichée.
  - (d) **Filtrage couleur** Le logiciel doit permettre de mettre en oeuvre les traitements couleur vus en TP, à savoir modification de la teinte (niveaux de gris, sépia, ...) ainsi que la sélection d'une teinte à conserver lors du passage en niveaux de gris. Chacun de ces traitements d'image possède son propre menu dans lequel on obtient la transformation voulue en appuyant sur un bouton.
  - (e) **Convolution** Le logiciel doit permettre d'appliquer différents filtres basés sur une convolution sur l'image affichée. Les filtres moyennneur, Gaussien, Sobel et Laplacien seront implémentés. Ces filtres sont ainsi implémentés dans le même menu pour la première release, puis seront séparés afin d'offrir une expérience plus agréable pour l'utilisateur.
6. **Réinitialiser** Le logiciel doit permettre de réinitialiser l'image c'est-à-dire d'annuler les effets appliqués depuis le chargement de l'image. Ainsi, dans chaque menu de transformation d'image se trouve un bouton qui remplace l'image modifiée par l'image originale, en la recopiant.

7. **Sauvegarder une image** Le logiciel doit permettre de sauvegarder une image modifiée. L'image est ainsi écrite dans la galerie dès que l'utilisateur décide d'appuyer sur le bouton de sauvegarde. Les développeurs laisseront au soin de l'utilisateur la gestion des différentes modifications de l'image originale : chaque modification pourra ainsi être enregistrée dans la galerie et quand l'utilisateur les jugera suffisantes, il pourra supprimer les images superflues directement dans la galerie.

## 2.2 Au choix

8. **Simuler un effet dessin au crayon** Le logiciel doit permettre à partir d'une photo de générer un dessin au crayon représentant l'image initiale. Cette fonctionnalité sera implémentée lors de la seconde version.
9. **Simuler un effet cartoon** Le logiciel doit restreindre les couleurs d'une image et renforcer les contours afin de lui donner un effet de bande dessinée. Cette fonctionnalité sera implémentée lors de la seconde version.
10. **Restreindre la zone d'application d'un filtre avec le doigt** Le logiciel doit permettre de restreindre l'application d'un filtre uniquement autour d'une zone définie par l'utilisateur avec son doigt. Cette fonctionnalité sera implémentée lors de la seconde version.
11. **Changer l'orientation de l'image** Le logiciel doit permettre la rotation de l'image dans le sens horaire ou dans le sens opposé. Cette fonctionnalité sera implémentée lors de la seconde version.
12. **Rogner l'image** Le logiciel doit permettre à l'utilisateur de rogner l'image en choisissant la taille des bords à rogner. Deux zones d'écriture seront ainsi disponibles pour que l'utilisateur puissent saisir le nombre de pixels ou de centimètres à retirer à droite, à gauche, en haut ou en bas. Cette fonctionnalité sera implémentée lors de la seconde version.

## 3 Architecture

Pour la première release, l'architecture du logiciel sera celle implémentée sur le premier schéma (voir Annexes).

Dans un souci de rendre le logiciel fonctionnel dès la première release, l'optimisation du code sera une des priorités de la seconde release. Ainsi, l'architecture de l'application tendra à ressembler à celle décrite par le seconde schéma (voir Annexes).

Il y aura également une deuxième release du cahier des charges afin de détailler plus en profondeur l'architecture proposée tout au long du développement.

## 4 Versions

### 4.1 Gestion

Le développement utilise l'outil de gestion de version *git*. Ainsi, l'application est hébergée sur un serveur *Github* : <https://github.com/cauli33/PhotoShip.git> .

### 4.2 Accès

L'accès au logiciel sera facilité par la mise en place de deux versions : une version française et une version anglaise. Celles-ci pourront être choisies par l'utilisateur au niveau du menu d'accueil de l'application.

### 4.3 Calendrier des mises à jour

Le projet démarrera le 20 Janvier, date de remise du sujet, et durera 3 mois. Il sera composé de deux releases du logiciel, qui correspondront aux dates de rendu de l'application.

1. **1er rendu - 3 Mars** La première mise à jour contiendra une première release du logiciel ainsi que le cahier des charges complété.

La première version du logiciel comportera tous les besoins fonctionnels de la section ??, ainsi que des ébauches du développement des fonctionnalités supplémentaires.

Par ailleurs, cette nouvelle version du cahier des charges contiendra la description des besoins complétés, l'architecture logicielle retenue détaillée et expliquée, ainsi que les choix de fonctionnalités supplémentaires qui semblent réalisables dans le temps restant.

2. **Rendu final - 14 Avril** La seconde release contiendra le logiciel terminé avec toutes les fonctionnalités énoncées précédemment. Une série de tests garantissant le bon fonctionnement du logiciel sera également fournie, ainsi qu'une évaluation des performances en temps et en mémoire.

### 4.4 Matériel utilisé

Les développeurs de ce logiciel ont utilisé les outils suivants :

1. **Android Studio** pour le développement de l'application
2. **Emacs et LaTeX** pour la rédaction de ce cahier des charges
3. **StarUML** pour la conception des deux diagrammes proposés

### 4.5 Contact

Les développeurs de ce logiciel peuvent être joignables par e-mail ou via leurs profils github :

1. Emma BESSE : [emma.besse@etu.u-bordeaux.fr](mailto:emma.besse@etu.u-bordeaux.fr) ou
2. Cauli HONORÉ : [quentin.honore@etu.u-bordeaux.fr](mailto:quentin.honore@etu.u-bordeaux.fr) ou <https://github.com/cauli33>
3. Emma VEILLAT : [emma.veillat@u-bordeaux.fr](mailto:emma.veillat@u-bordeaux.fr) ou <https://github.com/EmmaVeillat>



