Rapport de projet
Retouche d'images sur Android

Manuel Ricardo Guevara Garban
Loïc Lachiver
Romain Pigret-Cadou
Sofiane Menadjlia

L3 Informatique

Projet technologique

2020

Table des matières

1 Introduction :

TODO

2 L’application :

2.1 Lancement :

TODO

2.2 Utilisation :

TODO

3 Effets :

Interface :

L'utilisateur choisit un effet parmi la liste d'effets affichée en bas de l'écran. Il a ensuite accès aux réglages disponibles, s'il y en a. L'utilisateur peut ensuite choisir de confirmer ou d'annuler la modification apportée par l'effet choisi.

Structure du code :

Les effets sont rassemblés dans le package filters (fr.ubordeaux.pimp.filters). La classe *Retouching* contient les réglages de luminosité, contraste, saturation et teinte. La classe *Convolution* contient tous les effets liés à la convolution (flou, détection de contour etc.). Toutes ces méthodes sont appelées lors de l'appui de boutons ou de glissement de seekbars.

3.1 Luminosité (Brightness) :

3.2 Contraste (Contrast) :

3.3 Saturation (Saturation) :

4 Structure du projet :

4.1 Structure graphique Android et navigation :

Pour afficher certains éléments d'interface comme par exemple les informations de l'image (bouton ℹ) nous utilisons des **Fragment**. En effet une activité supplémentaire n'est pas nécessaire car cette petite partie de l'application ne correspond pas à un point d'entrée de l'application. Par ailleurs changer de fragment (plutôt que de changer directement de layout) pourrait faciliter l'implémentation d'une interface différente, pour tablette par exemple.
On notera que dans la structure actuelle de l'application, l'image actuellement éditée est contenue et manipulée depuis l'activité principale. Les fragments n'apportent à l'application que des éléments d'interface.

4.2 Classe Image :

Cette classe a été conçu comme une alternative à l'utilisation directe de la classe **Bitmap** fournie par Android.
Le coeur de la classe est évidement une instance de Bitmap, qu'il est possible de récupérer à tout moment. Par ailleurs la classe offre des fonctionnalités supplémentaires, parmi celle ci notamment la possibilité de restaurer l'image à son état au moment de sa création ou de son chargement, via la méthode **reset()**.
On notera la nécessité pour Image d'avoir la référence d'une Activité de l'application, en effet elle est requise à plusieurs moments par les librairies Android pour charger la Bitmap en mémoire.

4.2.1 Classe ImageInfo :

La classe Image 4.2 génère et garde une instance de la classe **ImageInfo**, cette classe contient un grand nombre de valeurs à propos de l'Image (dimensions, coordonnées GPS, date de prise de vue, ...).
L'idée de cette classe est d'empaqueter toutes ces informations afin de faciliter le passage de ces informations à l'avenir, notamment à travers des Fragments ou des Activités. On notera que tous les accesseurs appliquent des opérations de formatage sur ces données, certaines opérations pourraient être déplacées dans les constructeurs s'ils venaient à être utilisés régulièrement.

4.3 Package util :

Ce package contient de nombreuses classes contenant des méthodes statiques permettant une meilleur factorisation du code.
La classe **Utils** offre par exemple des méthodes pour récupérer la taille de l'écran ou pour calculer un ratio de redimensionnement.
La classe **BitmapIO** permet d'effectuer le chargement d'une Bitmap de plusieurs manières, depuis les resources ou un autre emplacement du téléphone, et avec la taille voulue.

5 Performances :

5.1 Temps d'exécution :

TODO

5.2 Mémoire :

TODO montrer l'ancienne instance d'image se faire garbage collecter ? voir remarques
TODO

6 Remarques et améliorations :

6.1 Remarques sur le code

TODO Remarques poids mémoire de la copie orginale des Images.

Lors du chargement d'une nouvelle image, nous ré-instancions un objet de la classe Image. Ce qui veut techniquement dire que jusqu'au prochain passage du ramasse miette Android, deux images sont en mémoires, donc deux Bitmap et deux tableaux de pixels (la copie originale des Images, voir 4.2). C'est un élément discutable cependant notre application limite la taille des Images chargées. Ce qui évite largement les dépassements mémoire.

6.2 Remarques sur les librairies Android

TODO Remarques sur l'utilisation obligatoire d'une activité contexte pour charger une Bitmap.

6.3 Améliorations à court terme :

TODO

7 Conclusion :

TODO

8 Annexes :

TODO