

IMAC 1 - Projet Algorithmique en C- 2015-2016
Rapport

MINIGimp

VINCENT Steeve
HADDAD Joya

Sommaire

[Sommaire](#)

[Introduction](#)

[Analyse fonctionnelle generale](#)

[IHM](#)

[Imagimp](#)

[Composant](#)

[Dialogue](#)

[Clavier](#)

[Image](#)

[Calque](#)

[Analyse fonctionnelle détaillée](#)

[Fichier et répartitions](#)

[Fonctionnement principal](#)

[Image PPM](#)

[Calques](#)

[Histogrammes](#)

[Traitement d'image](#)

[IHM](#)

[Fonctionnalité d'imagimp](#)

[Clavier](#)

[Boutons](#)

[Annexes](#)

Introduction

Ce projet intitulé Imagimp fut l'occasion de programmer un logiciel de traitement d'images tel que Photoshop ou Gimp (mais de plus petite envergure). Après avoir analysé le sujet et après quelques recherches sur les notions requises pour créer un logiciel de traitement d'image, nous avons procédé étape par étape pour créer notre Imagimp. En allant du général au particulier, nous allons commencer par présenter le fonctionnement général du programme, pour ensuite s'attarder sur le détail des éléments importants codes. Nous évoquerons également les difficultés rencontrées ainsi que les méthodes employées pour les surmonter.

1. Analyse fonctionnelle générale

1.1. IHM

Interface graphique fournie: glimagimp

La librairie fournie est une surcouche de glut, elle permet dans un premier lieu de faire l'affichage de l'image. A cette librairie, nous avons ajouté d'autres fonctionnalités afin d'obtenir une IHM.

Nous avons notamment rajouté des structures de données permettant de faire des interactions avec l'utilisateur.

1.2. Imagimp

La structure mère se nomme Imagimp, il s'agit d'une structure avec instantiation unique (Il n'existe qu'un seul objet de ce type), elle possède toutes les informations liées au bon fonctionnement de l'IHM. Elle possède notamment l'image de rendu composé de tous les calques. Ainsi que différentes variables informant de l'état du programme. Enfin on y retrouve la liste des composantes fixes.

En utilisant les différents événements gérés par glimagimp (auxquels nous avons rajouté celui du mouvement de la souris), nous avons implémenté les interactions entre l'utilisateur et les différents composant d'Imagimp.

1.3. Composant

La structure principale est Component, elle peut ensuite se décliner à l'aide d'une union afin d'obtenir plusieurs composants avec des caractéristiques communes. Cela nous permet d'avoir une liste de composants et de faire appel au même traitement sur chaque composant. Ces traitements font ensuite appel aux traitements spécifiques selon le type de composant que l'on manipule.

On a ainsi créé des sliders, des boutons, des boutons radios. Le type checkButton est aussi présent mais n'a pas été testé ni utilisé par manque de temps.

Chaque instance du type Composant possède un pointeur de fonction qui est appelé au moment d'un clique. Selon l'instance et le type de composant les actions engendrés seront différents. Dans beaucoup de cas, une fenêtre de dialogue est ouverte.

1.4. Dialogue

La structure Dialogue comme la structure mère instancie un seul objet utilisable dans tout le programme. Elle est modulable, c'est à dire qu'on peut la façonner avec des aspects différents. L'appel du Dialogue est accompagné de "drapeaux", des valeurs constantes, qui vont définir ce qu'on le va retrouver dans le Dialogue. Il peut être utilisé pour faire des saisies, pour définir une valeur à l'aide d'un slider, pour sélectionner un mode ou pour afficher des boutons qui ne peuvent être affichés directement dans l'IHM.

1.5. Clavier

Durant ce projet, nous avons implémenter l'IHM à l'instar de la gestion du clavier. Les commandes possibles sont assez restreintes, elle sont utilisées pour changer de calque courant (tab et maj+tab) et pour zoomer (ctrl->+ et ctrl->-). Les modifieurs (alt,ctrl) furent un problème durant ce projet, l'utilisateur ne peut pas appuyer sur un modifier et une autre touche en même temps. Le problème n'a pas été résolu, la solution a été substituée par la possibilité de zoomer en appuyant sur ctrl puis sur la touche +.

1.6. Image

L'image, décrite dans la structure Picture, est une liste chaînée double de calque. Elle est utile pour pouvoir naviguer entre les calques, on évite de naviguer tous les calques afin de retrouver le calque d'en-dessous. Le nombre de calque est accessible dans cette structure. L'image possède un calque blanc qui est utilisé comme base et ne peut être ni déplacé ni supprimé.

Un pointeur vers le dernier calque permet d'ajouter rapidement un nouveau calque à la fin. Un pointeur vers le calque courant donne accès au calque qui va recevoir les potentiels modifications, il est accompagné de son id.

1.7. Calque

Un calque décrit une et seule image caractérisée par un tableau 1D d'octets non signé, par sa largeur et sa hauteur. Chaque calque va être fusionner au calque précédent selon le mode de mélange qu'il définit. Avant d'être mélangé, le calque va être modifié selon une liste de LUT (Look at Table). Enfin un calque possède 4 histogrammes donnant des informations sur sa colorimétrie.

2. Analyse fonctionnelle détaillée

2.1. Fichier et répartitions

Main.c: lance la fonction principale

Interface.h + Interface.c : librairie fournis complétée avec les composants de l'IHM

Outils.h + outils.c : fichier avec des fonctions utilitaires et des fonctions de dessins

Layer.h + Layer.c : Gestion des calques et des histogrammes

Blend.h + Blend.c : Gestion des melanges entre calques

Imagimp_dialog.h + Imagimp_dialog.c : Gestion de la fenetre de dialogue

Imagimp_string.h + Imagimp_dialog.c : Gestion de chaine de caractère sous forme de liste chaînée.

Picture.h + Picture.c : Gestion d'une image : composition de plusieurs calque

Imagimp.h + Imagimp.c : Gestion principale du programme : événement, affichage, gestion de l'image principal

Ppm.h + ppm.c : lecture et ecriture de fichier PPM

2.2. Fonctionnement principal

2.2.1. Image PPM

Le format PPM est un format d'images reconnu par divers éditeurs d'images.

Le fichier PPM definis :

- La largeur et la hauteur
- La valeur maximun
- Le format d'ecriture

L'ecriture (et donc la lecture) des pixels dépend du format :

- P5 : Un pixel est un niveau de gris codé sur un octet non signé (0 à 255)
- P6 : Un pixel est une combinaison de trois canaux (rouge, vert, bleu) codée sur trois octet non signé (0 à 255 pour chaque canaux)

Les autres formats ne sont pas prise en compte dans ce projet

L'ensemble des pixels est décrit sur une même ligne.

Si le format est P5, il faut lire largeur x hauteur pixels, sinon il faut lire largeur x hauteur x 3 pixels.

2.2.2. Calques

Il a fallu créer des fonctions pour "créer" un calque,

Des fonctions pour mettre des calques en "arriere", et si ces derniers ont été "blend", si on veut le remelanger, ça mélange la dernière version finalisée avec le nouveau calque. On peut également enlever le calque de l'image avec la fonction free layer.

2.2.3. Histogrammes

Créer les histogrammes RGB, mettre à jour l'histogramme et enfin le dessiner l'histogramme.

2.2.4. Traitement d'image

— LUT : Contraste, Luminosité, Sepia, Négatif; Noir et Blanc, + affiche Liste

La modification d'un histogramme est représentée sur une courbe (appelée *courbe tonale*) indiquant la modification globale des composantes de l'image avec en abscisse les valeurs initiales et en ordonnées les valeurs après modification.

La courbe tonale correspond à une fonction de transfert définie par une table de transcodage appelé *look up table*, notée *LUT*. Ainsi la diagonale indique la courbe telle que les valeurs initiales sont égales aux valeurs finales, c'est-à-dire lorsque aucune modification n'a eu lieu. Les valeurs à gauche de la valeur moyenne sur l'axe des abscisses représentent les pixels "clairs" tandis que ceux à droite représentent les pixels foncés. Les LUT ont été donc utilisés pour la luminance, le contraste, sepia

— Melanges

Concernant l'image final, pour faciliter son rendu, un calque de rendu partiel est aussi ajouté à l'image, il s'agit d'un calque qui rend tous les calques en dessous du calque courant étant donné qu'il s'agit d'un rendu qui ne dépend pas de ce calque.

Et à chaque modification du calque courant, on rend l'image final en partant du calque partiel.

2.2.5. IHM

La structure mère des composantes est une structure possédant d'une part les informations commune à tous les composants et d'autre part une "union" permettant de réunir différentes caractéristiques spécifiques aux composants.

On retrouve principalement dans les informations communes, la zone où se trouve le composant ainsi que la fonction qui doit être appelée lorsque l'événement clique de la souris s'est enclenché dans la zone en question.

Parmi les informations spécifique, on retrouve par exemple l'intitulé (une chaîne de caractère) que l'on retrouve dans les boutons ou la position du curseur que l'on retrouve dans les sliders.

C'est Imagimp qui va créer tout les boutons nécessaires et en gérer les événements. Lorsqu'il reçoit un événement de la souris, il va aller chercher dans sa liste de composante et va enclencher le bon comportement du bon composant.

Le dialogue est un ensemble de composant qui va s'afficher ou non selon ce qu'on lui passe en paramètre. Le paramètre est décomposé bit par bit et selon les bits activés, le dialogue va rendre invisible ou non un composant. Il demande aussi en paramètre une fonction qui va être appelée lorsque le dialogue se ferme. Le dialogue renverra la réponse de l'utilisateur via cette fonction qui aura pour but de gérer la réponse.

Imagimp gère aussi le zoom ainsi que la translation des images. En revanche on ne peut pas redimensionner une image.

3. Fonctionnalité d'imagimp

3.1.1. Clavier

'v' : changement d'affichage, rendu -> calque original ou calque original -> rendu

'b' : changer de calque courant -> calque d'en dessous

'n' : changer de calque courant -> calque d'au dessus

'q' : quitter Imagimp

'x' : supprimer le calque courant

'm' : ouvrir le dialogue de mélange

'haut, bas, droite ou gauche' : effectue une translation de l'image ou change la valeur du slider courant

3.1.2. Boutons

Charger : ouvre une fenetre de dialogue, entrez le chemin du fichier pour l'afficher. Vous pouvez utiliser un chemin relatif, le chemin débute à partir du dossier d'exécution.

Sauvegarder: ouvre une fenetre de dialogue, entrez le chemin du fichier pour l'enregistrer. Vous pouvez utiliser un chemin relatif, le chemin débute à partir du dossier d'exécution.

Quitter : quitter le programme

Afficher calque original/ rendu: affiche le calque courant sans les modifications apporté/ le rendu final

Histogramme Rouge/Vert/Bleu/Luminosité/Tous: Affiche le ou les histogrammes du rendu final

Redimensionner: ouvre une fenetre de dialogue, entrez les dimensions du rendu final sous la forme (largeurXhauteur)

Avancer/Reculer: faire reculer le calque courant

Ajouter LUT : affiche une fenetre de dialogue affichant la liste des LUTs applicables, vous pouvez naviguer entre les boutons.

Liste LUT : affiche une fenetre de dialogue affichant la liste des LUTs du calque courant. Vous pouvez en supprimer en cliquant dessus.

Opacite: affiche une fenetre de dialogue avec un slider qui permet de gérer l'opacité du calque courant

Melange : affiche une fenetre de dialogue avec une liste de mode de melange

4. Annexes

