

Nom: TCHEWO YETENDJE HARDING JUNIOR

Matricule: 000604511

Rapport d'Analyse de Code

Introduction

Photobench est une application que nous avons développée en langage Processing dans le cadre de notre travail personnel pour le compte de la matière STICB450 Algorithmes et programmation I. Il s'agit d'un outil graphique interactif permettant de manipuler des images à travers une interface utilisateur. Le code inclut des fonctionnalités variées, notamment la gestion des outils de sélection, l'application de filtres Blur, Invert et GrayScale a une image, mobilité de l'images, d'un zoom avant et arrière, l'enregistrement de l'image dans un format jpg et un système d'historique pour effectuer des opérations d'Undo/Redo. Le but principal de ce programme est d'offrir une interface intuitive pour des opérations de base sur des images. Ce rapport se concentre donc sur la description de la structure, les fonctionnalités clés et les difficultés rencontré lors de la création de notre code pour l'application photobench.

1. Structure du Code

1.1. Variables Globales

Notre code déclare un ensemble de variables globales qui structurent notre application photobench. Ces variables ont pour rôle principale d'augmenter la lisibilité et la modularité du code mais ils permettent aussi une meilleure maintenabilité pour plus tard, ce qui augmente son efficacité. Nous avons donc comme variable :

- **Images :** `PImage` img, icon0, icon1, ..., icon7` : Utilisées pour afficher des images et des icônes dans l'interface utilisateur.
- **Position et Taille :**
 - `Float` imageLargeur, imageHauteur` : Variables pour gérer les dimensions des images chargées.
 - `Float` positionImgX = 150, positionImgY = 0 : Définit la position de l'image principale au lancement de l'application
 - `Float` zoomFactor = 1.0 : pour gérer le zoom avant et zoom arrière
- **Interactions Utilisateur :**
 - `String` NomDuboutonActif : Identifie l'outil actuellement sélectionné.
 - `Boolean` isDragging : Détecte si une action de Drag(glisser) de la souris est en cours.
 - `Boolean` imgBouge : Détecte si l'image entière bouge
 - `Int` x1, y1, x2, y2` : Coordonnées de délimitation d'une zone de sélection.
- **Historique :**

- **ArrayList** : Utilisé pour stocker les états successifs de l'image, permettant un système d'Undo/Redo.

1.2. Structure Fonctionnelle

- La gestion des événements de l'utilisateur (interactions souris, avec le clavier, sélection d'outils).
- La manipulation directe des images (Zone de sélection sur l'image, déplacement image, ajout des filtres sur l'image).
- Une interface utilisateur basée sur des boutons a icônes.

2. Fonctionnalités Clés

• Système d'application de Filtre et de drag de l'image

- String NomDuboutonActif : elle est utilisée pouvoir appliquer des filtres à et bouger l'image. Quand l'un des boutons de filtre ou le bouton Drag sera cliqué, la variable NomDuboutonActif sera passé en paramètre a la fonction **appliquerFiltre(String tool)** pour pouvoir appliquer le filtre approprié.

• Système de Gestion de Mémoire

- L'utilisation d'un ArrayList : pour gérer l'historique des états permet d'implémenter efficacement les opérations d'Undo et Redo.

- Chaque modification d'image sera donc sauvegardée à chaque fois que la fonction **appliquerFiltre(String tool)** sera appelé, offrant une flexibilité pour pouvoir faire un redo et un undo.

• Interactions avec la Souris

- Le booléen isDragging : indique une fonctionnalité permettant de dessiner une zone de sélection sur l'image, elle est mise a « true » quand le bouton select est pressé et revient à false lorsqu'on relâche, mais la zone de sélection se forme uniquement lorsque la souris est sur la zone réservé a l'image.

- Les coordonnées x1, y1, x2, y2 permettent de définir une zone rectangulaire, ces variables sont utilisées pour mémoriser la position initiale et les positions sur x et y lorsqu'on bouge la souris jusqu'à la position finale. Ces coordonnées seront également utilisées pour la création d'un rectangle qui s'adapte au mouvement de la zone de sélection.

• Affichage Graphique

-**barre de tâche** : qui contient les boutons

-**zone de l'image** : où sera mis l'image

-**Une image** : c'est sur cette image que toutes les modifications seront faites.

-**boutons** : nous avons créé 12 boutons qui contiennent des icônes pour interagir avec l'image.

3. Difficultés Rencontrées

La principale difficulté rencontrée a été la création d'une zone de sélection circulaire. Je n'ai pas pu l'intégrer à mon code car à chaque fois que j'ai essayé de le faire les filtres ne s'appliquait plus sur mon image j'avais effectivement le tracé d'une zone circulaire de couleur rouge, mais quand j'essayais d'appliquer un filtre ça ne fonctionnait pas ; Et même la zone de sélection rectangulaire elle aussi ne fonctionnait plus. Du coup j'ai dû la supprimer.

Conclusion

A travers ce travail, j'ai pu m'exercer pleinement dans la manipulation du langage avec processing. De nombreuses notions floues ont pris tous leur sens grâce à ce projet notamment la notion de variable qui facilite la modularité surtout pour un projet de cette envergure. En fin de compte j'ai effectivement respecté la demande minimale du travail, j'ai ajouté les options basiques et les options à difficultés moyenne. L'option last mile m'a été inaccessible malheureusement.