

Rapport Projet

Clément LEVEQUE

8 Janvier 2023

Table des matières

1	Introduction	3
1.1	Contexte	3
1.2	Sujet	3
2	Problèmes et Solutions	4
3	Conclusion	6

1 Introduction

1.1 Contexte

Le but de ce projet est ici de créer un interpréteur, qui va exécuter un programme dont le code sera une image en deux dimensions. Cette image sera parcourue par l'interpréteur qui effectuera différentes actions selon l'image.

1.2 Sujet

Les images sont constituées de différents blocs de pixels contigus de même couleur. Ces couleurs sont séparées en trois catégories : les couleurs codantes qui sont au nombre de 18 ; les couleurs passantes, des couleurs non codantes dont la luminance est supérieure à 128 ; et les couleurs bloquantes, des couleurs non codantes donc la luminance est inférieure stricte à 128.

L'interpréteur va parcourir l'image en passant de blocs en blocs en suivant une direction et un bord. Il va d'abord chercher la frontière du bloc dans la direction actuelle, puis trouver le pixel le plus du bord actuel sur cette frontière. A partir de ce pixel il va trouver le nouveau bloc en fonction de la couleur du prochain pixel dans la direction actuelle. Si la couleur est codante alors le nouveau bloc est celui de cette couleur. Sinon en fonction de si cette couleur est bloquante ou passante on change le bord et/ou la direction puis on recommence.

Lorsque l'interpréteur passe d'un bloc à un autre, il effectue une action sur une pile initialement vide en fonction des différences sur les couleurs des deux blocs.

2 Problèmes et Solutions

Le premier problème a été rencontré lors de l'implémentation des fonctions direction et bord se situant dans *pile.c*. Il a fallut déterminer un moyen d'avoir une direction et un bord, ainsi les types direction *dir* et bord *bor* furent introduits, ayant un entier nommé *type* ainsi qu'un tableau de caractères. On initialise ces types avec deux fonctions *initdir* et *initbord* qui mettent l'entier *type* à 0, et les tableaux de caractères respectivement à *e, s, o, n* et *b, t*.

Ensuite, la fonction *tourne* a posé problème. Plus particulièrement le stockage des valeurs dépilées. Une autre pile au sein de la fonction a donc été définie afin de stocker les valeurs dans l'ordre, il a donc suffi de dépiler les valeurs pour les rempiler sur la pile initiale.

Puis, il a fallut trouver comment différencier les couleurs codantes des autres. Le type *codant* a donc été créé, possédant un caractère *couleur* et un autre *luminosite*. Avec la fonction *pixtocolor* implémentée dans *image.c*, la couleur du pixel était trouvée et un *codant* était retourné : *z, z* si il n'était pas codant, et si il l'était : *j, m, r, c, v, b* pour la couleur et *c, n, f* pour la luminosité.

Passons désormais aux fonctions implémentées dans *main.c*. Tout d'abord, la question s'est posée de comment déterminer le pixel recherché dans le bloc courant. Il fallait d'abord trouver la frontière puis aller au bord actuel de cette frontière. La fonction *newpix* a répondu à ce problème, en parcourant tout d'abord dans une direction tous les pixels afin de trouver celui dont la couleur correspondait et qui était dans le bloc actuel qui se trouvait le plus dans une direction donnée. Ensuite, elle fait de même en parcourant cette fois ci les pixels sur la frontière dans le sens du bord actuel. De cette manière, cette fonction modifie par effet les coordonnées du pixel recherché.

Il a ensuite fallut implémenter l'algorithme permettant de déterminer le nouveau bloc. Afin de ne pas avoir à réécrire toutes les disjonctions de cas suivant la direction, deux fonctions furent introduites : *coldir* renvoyant la couleur du pixel si on se déplace dans la direction actuelle en utilisant la fonction *pixtocolor*, et la fonction *pixdir* modifiant par effet les coordonnées si on se déplaçait dans la direction courante.

De même, l'algorithme si la couleur est bloquante étant utilisé à différents endroits, la fonction *bloquant* fut introduite.

Ensuite, un problème a été de savoir de quelle manière calculer les différences entre les blocs. Le type *diff* fut introduit, séparant les différences de couleurs avec l'entier *color* des différences de luminosité avec l'entier *brightness*. Pour

calculer les différences on a donc dû implémenter une fonction *difference* retournant un *diff*, où chaque couleur est associée à un entier. Il suffit donc de faire la différence entre les deux entiers pour la couleur et la luminosité.

3 Conclusion

Tout d'abord, le programme ne fonctionne pas correctement. En effet, dans le "meilleur" des cas, quelquechose est écrit sur la sortie standard, et le programme ne se finit pas comme c'est le cas pour *99bottles.ppm*.

Dans d'autres cas, le programme ne se termine jamais, c'est le cas pour *hw4.ppm*.

Puis dans le pire des cas, le programme renvoie une erreur (que je comprends mais dont la cause m'échappe) soit en ayant écrit sur la sortie standard auparavant, soit tout de suite. C'est le cas pour *hw3.ppm* et pour *hw4.ppm*.

De plus, le programme est en complexité n^2 , ce qui peut rendre l'exécution longue.