C-PAINT

INFO 1C

Romuald Hanriot – Aurélien Marc
DUT INFO | GROUPE C

C-PAINT

1. Préambule

Le projet que nous avons choisi est le projet Paint.
Ce projet nous intéressait dans le sens où derrière son air simple, se cache une vraie difficulté. Que cela soit pour la fonction de remplissage, que pour la fonction d'un polygone plein.
Notre objectif était de faire un Paint opérationnel dans plein de contexte. Se libérer au maximum des petites contraintes comme d'être obligé de taper le texte via la console. De devoir choisir la couleur puis l'outil ou l'inverse (avoir un ordre de sélection pour simplifier).

Il n'y a aucun bug relevé. Plusieurs cas de limite ont été un peu dur a réglé comme celui de faire en sorte que le crayon ne trace pas de trait aléatoire a son initialisation (pareil pour la gomme). Pareil pour le chargement de l'image. On peut changer n'importe qu'elle image en chemin relatif ou absolu mais au départ. Si l'image n'existait pas cela faisait crasher le logiciel. Maintenant cela est pris en compte et cela affiche un message d'erreur, fais un backup et ferme le logiciel.

Certain détail technique peut poser un souci de fluidité. On ne peut pas changer d'outil directement après avoir sélectionné un outil. Il faut d'abord cliquer une fois n'importe où puis recliquez sur le nouvel outil.

On avait imaginé des options supplémentaires comme par exemple : la visualisation en direct du placement des triangles, cercles et polygones. Mais très vite les limitations du C nous ont freiné.

2. Cahier des charges

L'une des premières étapes a été de créer un vrai cahier des charges et répartir la charge de travail entre les deux membres du binôme : voici le cahier des charges

C-paint	Groupe	Noms	Difficulté	Tâches finies
	OBLIGATOIRE	Segment	1	Oui
		Rectangle vide	1	Oui
		Cercle vide	1	Oui
		Polygone vide	1	Oui
		Rectangle plein	1	Oui
		Cercle Plein	1	Oui
		Texte de taille variable	1	Oui
		Gestion d'une palette d'au moins 8 couleurs	1	Oui
		Aide	1	Oui
		Charger image	1	Oui
	SU	Remplissage	3	Oui
			2	Oui

	PPLEMENTAIRES	Sauvegarde .bmp	3	Oui
		Undo	1	Oui
		Traitement de texte en temps réel	2	Oui
		Crayon/gomme	3	Oui
		Polygone plein	3	Oui
		Triangle	1	Oui
		Triangle plein	2	Oui
		Carré chromatique	1	Oui
		Pipette	1	Oui

3. Présentation des fichiers :

Global :

- global.h

Fichier contenant les define utilisé à plusieurs endroits et 4 variables globales. La couleur, la taille, la sauvegarde

Plateau :

- plateau.h

Fichier contenant les prototypes fonction s'appliquant au tableau tel que la vérification du plateau, la couleur, l'affichage des instructions

- plateau.c

Fichier qui utilise des fonctions pour le plateau, tout en utilisant les prototypes des fonctions de plateau.h et global.h pour éviter de surcharger le code.

Menu :

- menu.h

Fichier contenant le chemin de la lib graphique et de plateau.h

- menu.c

Fichier contenant le code du menu d'accueil pour savoir si on veut charger une image.

Main :

- main.c

Fichier contenant la barre des outils qui permet de savoir quel outil on veut choisir comme outil avec ses instructions

Couleurs :

- couleur.h

Fichier qui contient le prototype de la fonction sélection couleur

- couleur.c

Fichier qui contient la fonction de sélection des couleurs que peut utiliser l'utilisateur.

Lettre :

- lettre.h

Fichier qui contient le prototype de la fonction lettre

- lettre.c

Fichier qui utilise le prototype de la fonction lettre pour introduire des caractères que l'on peut utiliser avec la fonction d'écriture.

Outils :

- outil.h

Fichier contenant les prototypes des fonctions pour les outils

- outil.c

Fichier contenant toutes les fonctions permettant d'utiliser les outils du Paint

4. Conceptualisation

1ère étape :

Il y a plusieurs choses qu'on devait prendre en compte dans la conceptualisation du Paint.
Tout d'abord la gestion de plusieurs variables. Les variables de couleur, de l'épaisseur et de sauvegarde.

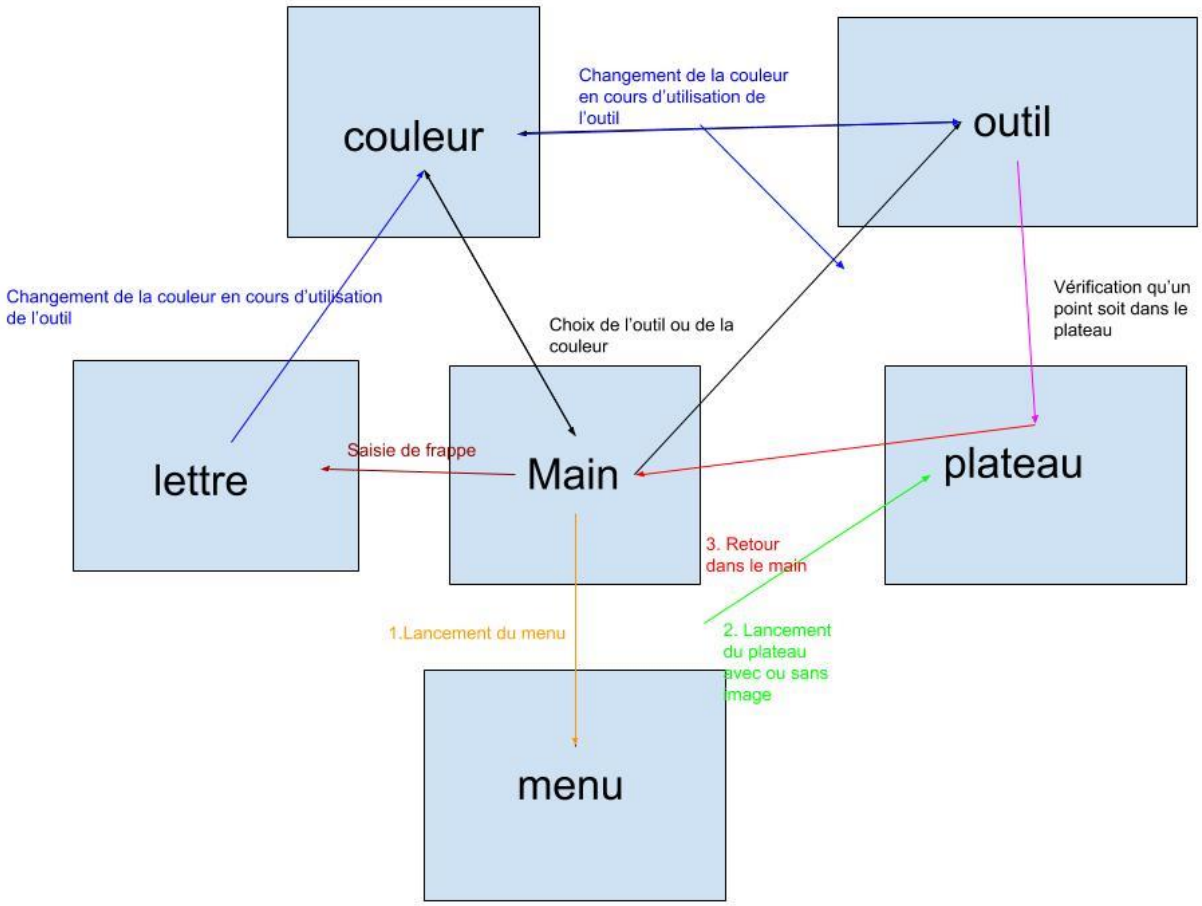
On souhaitait dans un premier temps ne pas avoir de variable globale, trouvant cela particulièrement. Malheureusement notre envie à naviguer librement à travers les outils, couleurs et taille nous a empêcher de faire cela

Les autres questions que nous nous sommes posées étaient des questions sur la structure de notre programme, nous sommes immédiatement partis sur plusieurs fichiers .c pour avoir des fonctions plus facilement retrouvables. La perceptive d'un seul fichier c ne nous a même pas traversé l'esprit.
Nous sommes partie sur un départ avec trois fichiers .c. le main, l'outil et le plateau.

2eme étape :

Pour pouvoir accomplir notre envie de naviguer librement parmis les options de notre programme et ayant reçu l'autorisation de Mr. Marsan, nous avons décidé d'utiliser 4 variables globales uniques : une pour la couleur choisis, une pour la couleur sur laquelle j'ai cliqué (crée initialement pour éviter les stacks overflow de la fonction de remplisage mais garder pars la suite, une pour la taille et une pour le tableau de sauvegarde.

Dans ce même temps nous avons décidé de faire un vrai schéma du fonctionnement de notre programme que voici ici présent :



REMARQUE :

Nous avions utilisé des variables globales alors que nous aurions pu utiliser des pointeurs. Malheureusement à l'heure de commencer le programme nous n'avions toujours pas vu les pointeurs. Et une vois vu tout cela est rester un peu trop flou pour nous.

En plus de cela nous étions beaucoup trop avancées et changer cela nous semblait beaucoup trop compliqué.

Nous vous prions de nous excuser de cela. Mais de ce fait nous avons quand même essayer d'utiliser le moins de variable global et nous l'avons fait que dans un cas juger utile et nécessaire.

5. Les points importants

Introduction :

Plusieurs points méritent d'être abordés plus précisément ; le crayon/gomme et le remplissage

Les outils importants :

- Le crayon

  Le crayon et la gomme nous ont demandé pas mal de temps avant d'être complètement implémentés.

  Dans un premier temps on pensait que faire un simple « *changer-pixel* » avec comme point qu'il reçoit les coordonnées de la souris suffirait. Mais cela ne marche pas dans le cas où on déplace vite son point. De ce fait on n'a pas mal fait de tests dessus. Avec des cercles ; rectangle et autres. Mais à chaque fois on retrouvait ces trous entre deux déplacements.
  Après pas mal de recherche on a trouvé quelque chose qui marchait à merveille : prendre un premier point, dessiner la ligner, prendre un deuxième point :
  Au premier passage cela ne fera rien. Le premier point sera en fait égal au deuxième point. Mais au deuxième passage le deuxième point et le premier point ne sont plus les mêmes donc cela permet de faire un dessiner ligne entre les deux. Pour grossir le trait on a juste utilisé de simple boucle qui dessinait plusieurs lignes à des coordonnées différentes.

- Le remplissage

  Le remplissage a été assez pénible à appréhender. L'algorithme classique ne marchait pas dans le sens où on avait un dépassement de pile. Nous avons du coup travaillé sur un autre algorithme :

  Faire de l'itérative complet était proscrit. Beaucoup trop long et prise de têtes. De ce fait on a diminué la récursivité de l'algorithme courant. Le but de la fonction sera de : vérifier que la couleur n'est pas celle qu'on souhaite

changer et est sur laquelle on a cliqué au départ. Puis ensuite on va utiliser un « *while* » à doubles conditions pour voir qu'elle est le point le plus gauche puis faire de même jusqu'à droite.
Ensuite, replacer le point le plus gauche et faire du récursive pour Y+ 1 et Y – 1 dans une boucle for qui va de tout à gauche à tout à droite.
Cet algorithme a un temps d'exécution assez rapide et demande moins de mémoire que l'algorithme de base.