

23/05/2023

Dessin vectoriel

PROJET EN C



Vénus BAKIKO et Corentin KERVAGORET
EFREI / Groupe F / L1 2022-2023

Table des matières :

I-Introduction.....	2
II-une présentation fonctionnelle du projet réalisé :.....	3
III- Une présentation technique du projet réalisé :.....	6
Conclusion :.....	8

I-Introduction

Dans le cadre du cours « Algorithme et structure de données », nous avons dû réaliser une nouvelle version d'un logiciel de dessin vectoriel.

En effet, le but de ce projet est de réaliser un logiciel de dessin vectoriel depuis la console. C'est-à-dire que l'utilisateur doit taper des instructions pour pouvoir dessiner des formes : cercle, point, ligne, rectangle, carré ou polygone.

Le projet a donc pour but de découvrir les possibilités du langage C et notamment comprendre le langage C, les notions de pointeur en C, les structures, et la gestion de la mémoire en générale.

Nous avons donc eu environ un semestre, de la date de présentation du sujet (le 18/03/2022) jusqu'à la date de soumission (le 22/05/2022), pour réaliser ce projet en binôme.

Ce projet est composé de deux étapes : l'étape de la programmation et l'étape finale de la présentation du projet.

Pour le fonctionnement du logiciel, nous avons donc utilisé 7 fichiers .h/.c (main, commands, area, pixel, formes, shape et id) ainsi que l'interface CLION pour pouvoir programmer.

II-une présentation fonctionnelle du projet réalisé :

Toutes les fonctions demandées dans la partie obligatoire ont été réalisées.

Les fonctions point et plot pour créer un point et afficher la matrice :

[illegible]

Les fonctions qui permettent de créer les différentes formes cercle, ligne, carré et rectangle :

[illegible]

[illegible]

Page 4 sur 9

```
>> Liste des formes :

1) POLYGON taille= 4, points= (2 2) (2 6) (6 6) (2 2)
2) POINT 1 1

3) POINT 2 6

4) CIRCLE (10 10),4

Entrez une commande :erase

>> Supression de toutes les formes !!

Entrez une commande :plot

* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
```

La commande help affiche toutes les commandes possibles:

```
Entrez une commande :help
- clear : effacer l'ecran
- exit : quitter le programme
- point x y : ajouter un point
- line x1 y1 x2 y2 : ajouter un segment reliant deux points (x1, y1) et (x2, y2)
- circle x y radius : ajouter un cercle de centre (x, y) et de rayon radius
- square x y length : ajouter un carre dont le coin superieur gauche est (x, y) et de cote length.
- rectangle x y width height : ajouter un rectangle dont le coin superieur gauche est (x, y), de
  largeur width et de longueur height
- polygon x1 y1 x2 y2 x3 y3 ... : ajouter un polygone avec la liste des points donnees
- plot : rafraichir l'ecran pour afficher toutes les formes geometriques de l'image (en fonction des
  regles d'affichage)
- list : afficher la liste de l'ensemble des formes geometriques qui composent l'image ainsi que
  toutes leurs informations
- delete id : supprimer une forme a partir de son identifiant id.
- erase : supprimer toutes les formes d'une image.
- help : afficher la liste des commandes ainsi qu'un mini mode d'emploi permettant a l'utilisateur
  d'utiliser les commandes correctement.
```

III- Une présentation technique du projet réalisé :

Nous avons divisé notre programme en 6 doubles fichiers (.c/.h) :

- Le fichier main → main.c
- Le fichier pour la création de la zone de dessin area → area.c
- Le fichier pour la saisie des commandes → commands.c
- Le fichier pour la création des formes → forme.c
- Le fichier pour la création du type générique shape → shape.c
- Et un fichier pour la création des pixels de chaque forme → pixels.c

Le fichier main est le cœur du programme, il permet de lancer le programme et coordonner toutes les fonctions nécessaires au bon fonctionnement du programme qui sont importés depuis les autres fichiers.

Tous les autres fichiers permettent de créer chacun des modules en fonction de son nom.

Ainsi forme.c permet de créer toutes les formes et shape.c les transforment en type générique pour pouvoir les utiliser dans pixel.c qui va calculer et allouer l'espace nécessaire pour la création de chaque forme en fonction du nombre de pixel.

Par rapport aux modifications apportées nous avons ajouté un paramètre à la fonction read_exe_command, on a ajouté le paramètre area pour qu'elle puisse être modifiée dans le programme.

Pour rendre notre code aussi plus performant nous avons ajouté des contrôles pour vérifier les données entrées par l'utilisateur. Nous avons donc pris en compte des erreurs de saisie :

Si l'utilisateur se trompe dans la saisie de la première commande :

```
Entrez une commande :squar
>> Commande inconnue
```

Si l'utilisateur ne donne pas assez d'information :

```
Entrez une commande :point 1
>> Valeur manquantes ou impossible
```

ou si un point est en dehors de la zone :

```
Entrez une commande :rectangle 12 35
>> Valeur manquantes ou impossible
```

Nous avons rencontré pas mal de problème tout du long de la réalisation de notre projet. Et l'un des problèmes majeurs a été le stockage de la mémoire. En effet il fallait allouer assez d'espace mémoire pour chaque forme notamment pour la ligne, le cercle et le polygone. Pour résoudre ce problème, affiché des messages pour voir où le programme s'arrêtait, était très

pratique. Pour résoudre le problème, on a, avant d'allouer le tableau de pixel, fait un tour de boucle vide de l'algorithme qu'on allait utiliser pour calculer le nombre de pixels.

Exemple avec la fonction circle :

Tour de boucle vide pour calculer le nombre de pixel :

```
// Calcul du nombre d'éléments pour allouer pixel
while (y >= x) {
    k=k+8;
    if (d >= 2 * x) {
        d -= 2 * x + 1;
        x++;
    }
    else if (d < 2 * (circle->radius - y)) {
        d += 2 * y - 1;
        y--;
    }
    else {
        d += 2 * (y - x - 1);
        y--;
        x++;
    }
}

*nb_pixels = k;
```


Programme avec la création des pixels :

```
while (y >= x) {
    //creation des 8 octans
    px = create_pixel( px: circle->center->pos_x + x, py: circle->center->pos_y + y);
    (*pixel)[k++] = px;
    px = create_pixel( px: circle->center->pos_x + y, py: circle->center->pos_y + x);
    (*pixel)[k++] = px;

    px = create_pixel( px: circle->center->pos_x + y, py: circle->center->pos_y - x);
    (*pixel)[k++] = px;
    px = create_pixel( px: circle->center->pos_x + x, py: circle->center->pos_y - y);
    (*pixel)[k++] = px;

    px = create_pixel( px: circle->center->pos_x - x, py: circle->center->pos_y - y);
    (*pixel)[k++] = px;

    px = create_pixel( px: circle->center->pos_x - y, py: circle->center->pos_y - x);
    (*pixel)[k++] = px;

    px = create_pixel( px: circle->center->pos_x - y, py: circle->center->pos_y + x);
    (*pixel)[k++] = px;

    px = create_pixel( px: circle->center->pos_x - x, py: circle->center->pos_y + y);
    (*pixel)[k++] = px;

    if (d >= 2 * x) {
        d -= 2 * x + 1;
        x++;
    }
    else if (d < 2 * (circle->radius - y)) {
        d += 2 * y - 1;
        y--;
    }
    else {
        d += 2 * (y - x - 1);
        y--;
        x++;
    }
}
```

Conclusion :

En conclusion, ce projet en C a été une expérience enrichissante et stimulante. A travers ce travail, nous avons pu développer encore plus nos compétences en programmation et approfondir notre compréhension des concepts clés en C. Par ailleurs, le projet nous a appris l'importance de la collaboration et de l'échange de connaissances. Nous avons pu travailler en équipe, partager nos idées et bénéficier des connaissances de chacun. Malgré quelques difficultés rencontrées dans le codage des fonctions, nous avons su surmonter les difficultés ensemble.

Dans l'ensemble, ce projet a été une expérience gratifiante qui nous a permis d'acquérir de nouvelles compétences et de développer des solutions concrètes, ainsi que de mieux comprendre la notion importante du C, notamment le stockage des variables en mémoire. Nous sommes donc fiers du travail accompli car nous avons réalisé le travail demandé. Nous sommes également confiants que les acquis nous seront utiles dans nos futurs projets.

