

Projet C : Vector Text-based Editor



Groupe BN

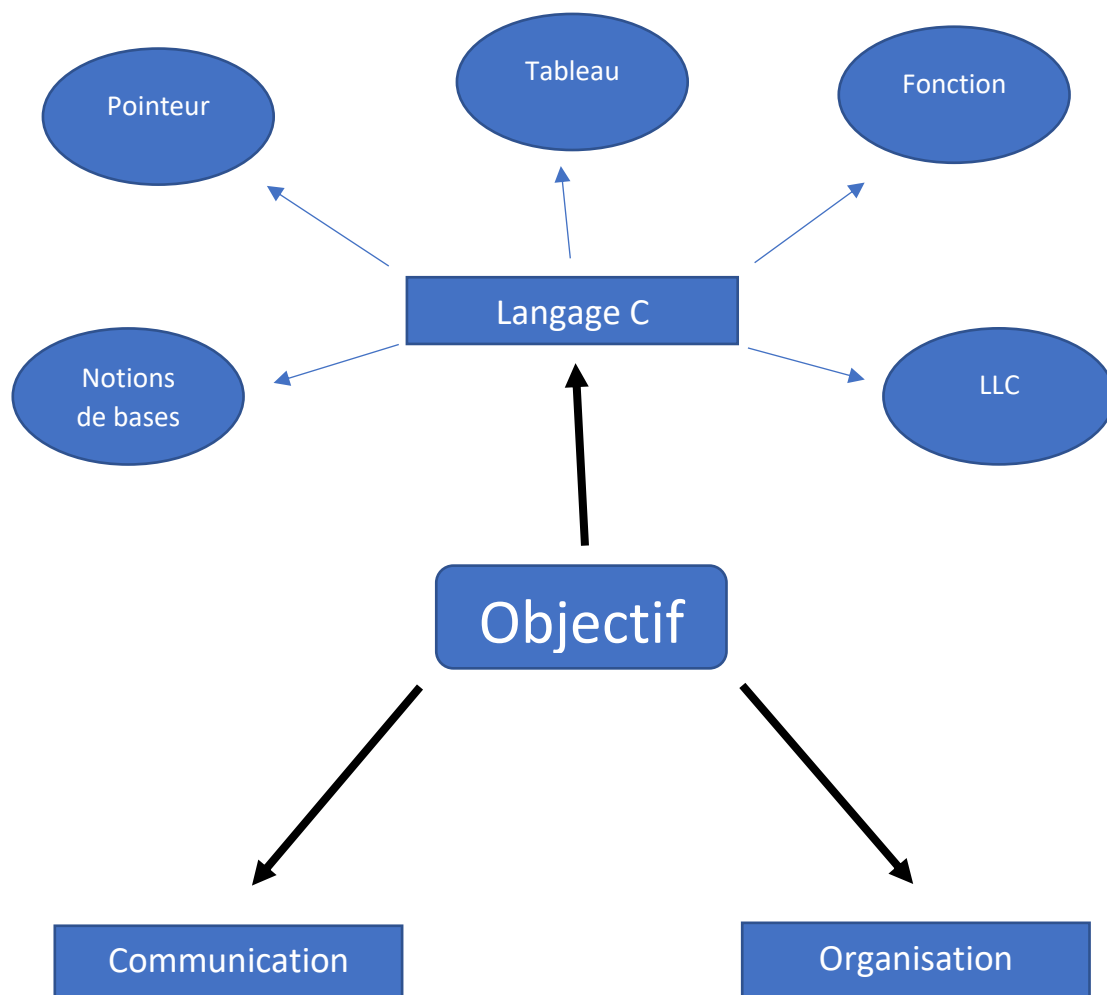
Hao Xuan Wilhem HO

Meitoh Joss Douniama

Introduction

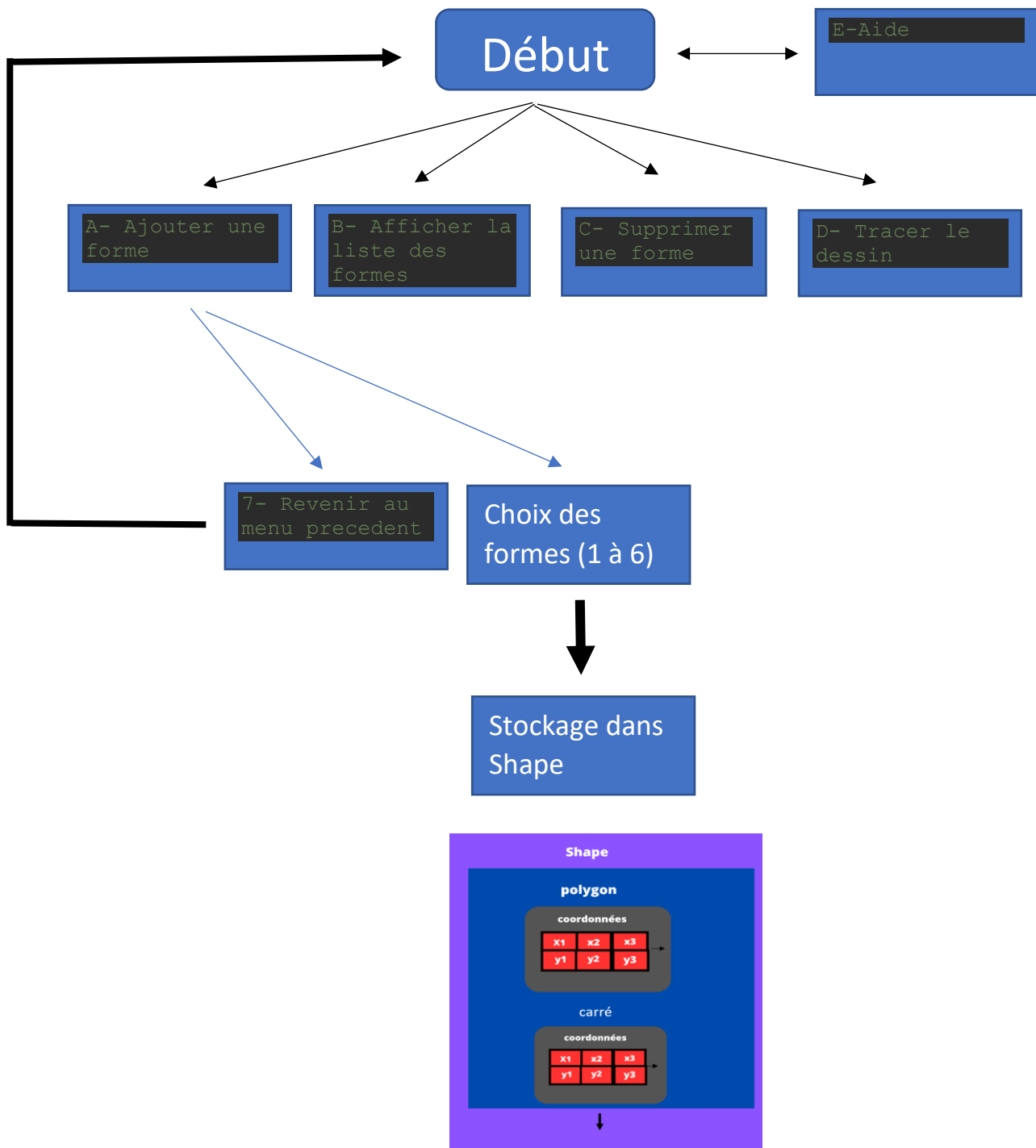
Dans ce projet, nous devons développer en langage C un programme de dessin vectoriel en mode texte, permettant à l'utilisateur de créer une image vectorielle simple. Le programme devra gérer une fenêtre, un calque regroupant les formes géométriques créées et offrir une interface en ligne de commandes pour effectuer diverses actions.

Dans la suite de ce projet, nous aborderons la représentation mémoire des informations, l'interface utilisateur en ligne de commandes, l'affichage des images vectorielles en mode texte, ainsi que d'autres fonctionnalités pour la création et la manipulation d'images vectorielles.



Présentation fonctionnelle du projet réalisé

Dans le menu, l'utilisateur peut choisir 4 fonctionnalités en saisissant une lettre de A à E correspondant aux fonctionnalités. Si l'utilisateur choisit d'ajouter une forme, alors il doit choisir les 6 formes proposées (point, ligne, carré, rectangle, cercle et polygone) et saisir ensuite les coordonnées de la forme qui sera stockée dans le tableau shape. Si l'utilisateur veut revenir au menu, il peut taper 7 pour revenir au menu.



Présentation technique du projet réalisé

Dans le code on a utilisé des boucles while pour que le code continu de fonctionner après avoir fini une action. Sans les boucles, le code va s'arrêter lorsque toutes les instructions sont exécutées. On a aussi utilisé des do...while pour la saisie sécurisée pour éviter à l'utilisateur de saisir d'autre chose et pour que l'utilisateur puisse ressaisie.

```
int main() {
    char x1,y1;
    int x ,y ,len,rad,largeur,px1,py1,px2, py2,cote;
    int t=1,verif,*tab;
    while (t==1) {
        do
        {
            printf( format: "Veuillez choisir une action :\n"
                "A- Ajouter une forme\n"
                "B- Afficher la liste des formes\n"
                "C- Supprimer une forme\n"
                "D- Tracer le dessin\n"
                "E- Aide\n"
                ">> Votre choix : ");
            scanf( format: " %c", &x1);
        }
        while (x1!='A' && x1!='B' && x1!='C' && x1!='D' && x1 !='E');
        verif=1;
        while (verif==1)
        {
            switch (x1) {
                case 'A':
                    do{
                        printf( format: "Veuillez choisir une forme :\n"
                            "1- Ajouter un point\n"
                            "2- Ajouter une ligne\n"
                            "3- Ajouter cercle\n"
                            "4- Ajouter un carre\n"
                            "5- Ajouter un rectangle\n"
                            "6- Ajouter un polygone\n"
                            "7- Revenir au menu precedent\n"
                            ">> Votre choix : ");
                        scanf( format: " %c", &y1);
                    }
```

Pour stocker les formes créer, on a créé une tableau Shape pour stocker la forme avec une id puis il va stocker les coordonnées (x y) saisie par l'utilisateur pour la forme. Voici les fonctions qui permet de à l'utilisateur de créer les 6 formes et stocker dans Shape :

```
#ifndef PROJET_VTE_SEMESTRE2_SHAPE_H
#define PROJET_VTE_SEMESTRE2_SHAPE_H

typedef enum { POINT, LINE, SQUARE, RECTANGLE, CIRCLE, POLYGON} SHAPE_TYPE;

typedef struct {
    int id; // identifiant unique de la forme
    SHAPE_TYPE shape_type; // type de la forme pointé
    void *ptrShape; // pointeur sur n'importe quelle forme
}Shape;

Shape *create_empty_shape(SHAPE_TYPE shape_type);
Shape *create_point_shape(int px, int py);
Shape *create_line_shape(int px1, int py1, int px2, int py2);
Shape *create_square_shape(int px, int py, int length);
Shape *create_rectangle_shape(int px, int py, int width, int height);
Shape *create_circle_shape(int px, int py, int radus);
Shape *create_polygon_shape(int lst[], int n);
void delete_shape(Shape * shape);
void print_shape(Shape * shape);

#endif //PROJET_VTE_SEMESTRE2_SHAPE_H
```

Présentation des résultats

Dans ce résultat, on a choisi la fonctionnalité D qui consiste à tracer le dessin dans la matrice. Ici la matrice est vite car on n'a pas encore ajouté de forme donc le tableau shape est vite

[illegible]

On peut supprimer les formes dans le tableau avec la fonctionnalité C

```
A- Ajouter une forme
B- Afficher la liste des formes
C- Supprimer une forme
D- Tracer le dessin
E- Aide
>> Votre choix :C
  Veuillez saisir l'ID de la forme a supprimer :1
  La forme avec l'ID 1 a ete supprimee.
Veuillez choisir une action :
A- Ajouter une forme
B- Afficher la liste des formes
C- Supprimer une forme
D- Tracer le dessin
E- Aide
```

Conclusion

Malgré on n'a pas terminer ce projet mais ce projet en C nous a permis de mieux maitriser la langage C (notions des bases, listes, les fonctions, ...), voir approfondir dans certaines notions en langage C en particulier en allocation dynamique du stockage. Lorsqu'on rencontre des problèmes, on effectue des recherches sur internet et parfois on trouve des solutions hors du cours. Cela nous permet d'apprendre des nouvelles techniques et solutions en autonomie pour régler certains problèmes en C. On n'a pas vraiment partagé les travaux au niveau du code, on a travaillé ensemble pour les codes car on estime travailler ensemble nous permet de mieux comprendre la totalité du code contrairement aux partages du travail

Le fait de travailler ensemble nous a permis d'être plus efficace, de mieux comprendre le code et de mieux gérer notre temps. La communication est très importante aussi dans ce projet car on communique beaucoup concernant certains points du projet