

Rapport sur le Projet FreeCell en Langage C

Réalisé par : Abbassi Zaid

1. Objectif du Projet :

L'objectif principal de ce projet était de créer une version du jeu FreeCell en utilisant le langage de programmation C et les structures de données (les piles).

Le jeu devait respecter les règles classiques de FreeCell et offrir une expérience utilisateur interactive en mode console.

2. La structure de donnée utilisé :

1-la structure :

```
typedef struct cartes {  
    int numero;  
    char forme;  
    struct cartes *next;  
} tCarte;
```

2- Pourquoi cette structure :

Dans cette structure utilisée on a, le numero qui représente le numéro de la carte, la forme de la carte que ça soit cœur, carré, trèfle ou pique et next qui représente l'élément suivant de la pile.

Vous avez peut-être une question pourquoi dans la structure il n'y a pas un élément qui représente la couleur. La réponse est simple, on n'a pas besoin car si on connaît la forme on connaît la couleur et c'est ça le principe que j'ai utilisé pour comparer la couleur dans les fonctions de mouvement.

3. Les zones de jeu :

Chaque zone de jeu est un tableau de structure d'adresse et la dimension change selon le tableau. Pour la zone 1 la dimension est 8 cases pour la zone 2 et 3 les deux sont de dimensions 4 au début du jeu chaque tableau est initialisé à NULL au début du jeu.

4. Fonctions Implémentées :

1- Initialisation du jeu :

1-1) void initialiserVariables(void) ;

Cette fonction initialise chaque case des tableaux de zone1, zone2, et zone3 à NULL.

2- Création et distribution aléatoire des cartes dans les piles :

2-1) void Creer_Carte(int pos, int numero, char forme) ;

Cette fonction crée une carte qui a un numéro et forme et l'insérer dans une position. Cette fonction sera appelée dans la fonction *genererJeuDeCartes*

2-2) void genererJeuDeCartes(void) ;

Cette fonction va créer des cartes aléatoirement et les insérer dans une position aléatoire en utilisant la fonction *Creer_Carte*. La position aléatoire sera générée en utilisant la bibliothèque "time.h" la fonction va arrêter quand on arrive à 52 cartes

2-3) void distribuerZone1(void) ;

À partir de la pile qui contient 52 cartes on va distribue dans la zone 1 dans les 4 premiers lignes la fonction distribue 4 cartes les 4 dernières lignes elle va distribue 4 cartes

3-Afficher les zones 1, 2 et 3 :

3-1) void afficher(void) ;

Cette fonction utilise les fonctions exclusives de la bibliothèque "windows.h"

GetStdHandle(STD_OUTPUT_HANDLE), GetConsoleScreenBufferInfo(hConsole, &consoleInfo) et SetConsoleTextAttribute(hConsole, attribute) pour contrôler la console et afficher en couleur

4-Déplacement des cartes selon les règles du jeu :

4-1) void de_Zone1_vers_Zone3(void) ;

4-2) void de_Zone1_vers_Zone2(void) ;

4-3) void de_Zone2_vers_Zone1(void) ;

4-4) void de_Zone2_vers_Zone3(void) ;

4-5) void de_Zone1_vers_Zone1(void) ;

Chacune de ces fonctions quand elle est appelée elle demande au joueur une source et une destination et vérifie si les règles du jeu sont respectées si oui le mouvement est effectué sinon il affiche une erreur.

5-Vérification de la victoire ou de la défaite du joueur :

5-1) bool verifieVictoire(void) ;

Vérifie si le nombre de la dernière carte de chaque case de la zone 3 égale à 13 si oui le joueur a gagné sinon pas encore.

5-2) int game_over(void) ;

À l'aide des fonctions is_blocked_1(), is_blocked_2(), is_blocked_3(), is_blocked_4 et is_blocked_5() cette fonction vérifie s'il y'a un mouvement possible si oui le joueur n'a pas encore perdu sinon il a perdu.

6-La fonction aide :

6-1) void help(void) ;

À l'aide des fonctions aide_1(), aide_2() et aide_3() s'il y'a un mouvement disponible la fonction affiche la ligne où le mouvement est possible.

5. Bibliothèques et Techniques Utilisées

Le développement du projet FreeCell a fait usage des bibliothèques suivantes et de techniques spécifiques :

3.1. Bibliothèque "head" :

Une bibliothèque locale nommée "head" a été créée pour contenir les prototypes des fonctions utilisées dans le projet. Cette approche modulaire a facilité la distinction entre les déclarations de fonctions et leurs implémentations, améliorant ainsi la clarté du code source.

3.2. Bibliothèque "windows.h" pour l'affichage en couleur :

La bibliothèque windows.h a été intégrée pour permettre l'affichage en couleur dans la console Windows.

Cette fonctionnalité a été utilisée pour améliorer l'aspect visuel du jeu FreeCell en utilisant des couleurs pour différencier les cartes et les piles, améliorant ainsi l'expérience utilisateur.

3.3. Fichier "impl.c" pour les implémentations :

Les implémentations complètes des fonctions du projet FreeCell ont été regroupées dans le fichier source "impl.c".

Ce fichier contient la logique détaillée pour la gestion des cartes, la distribution aléatoire, le déplacement des cartes, et les vérifications de victoire ou de défaite.

6. Difficultés Rencontrées :

Lors du développement du jeu, il y avait deux défis : mettre en œuvre les règles du jeu et l'affichage en couleur. Ces défis ont été surmontés par des recherches et des expérimentations

7. Tests Réalisés :

Des tests ont été effectués pour garantir le bon fonctionnement du jeu.

Cela inclut des tests de déplacement de cartes, des tests d'initialisation du jeu, et des tests de conditions de victoire.

Ces tests ont permis de garantir la qualité et la stabilité du jeu.

8. Conclusion :

En conclusion, ce projet a permis d'acquérir une expérience pratique dans le développement en langage C.

Il a également permis de renforcer la compréhension des structures de données et des algorithmes liés aux jeux de cartes.

Ce projet a été une opportunité enrichissante pour appliquer mes connaissances.