

RAPPORT DE PROJET

Dans le cadre de ce projet, nous allons chercher à suivre intégralement le processus d'un programme en C s'appuyant sur des listes doublement chaînées. L'objectif est d'avoir un programme fonctionnel, proposant les diverses fonctions imposées et ergonomique, avec une interface utilisateur convenable.

I – Portée du projet

L'objectif de ce projet est de créer un programme proposant 9 fonctions principales :

- 1- Afficher la liste des citoyens
- 2 - Afficher la liste des lieux de fêtes
- 3 - Ajouter un citoyen
- 4 - Ajouter un lieu de fêtes
- 5 - Remplir les participants a une fête
- 6 - Afficher tous ceux qui ont rencontré un citoyen
- 7 - Enregistrer un fichier situation
- 8 - Ouvrir un fichier situation
- 9 - Supprimer tous les évènements antérieur à une date

Pour répondre à cet objectif, il est imposé dans le cahier des charges d'utiliser deux listes doublement chaînées : « lieux » et « citoyen » ainsi qu'une simplement chaînée « participant ».

II- Conception du projet

Dans un premier temps j'ai voulu structurer les listes en créant les « bidons » et le chaînage de base à l'initialisation du programme main.

Par la suite, j'ai créé des fonctions d'ajout manuel (listés comme manuadd dans le programme) d'éléments dans les listes qui pourront plus tard être appelées dans les différentes fonctions du programme ou utilisées simplement dans le main a des fin de test.

Toujours dans cette optique de pouvoir tester le programme tout au court de son développement, j'ai par la suite intégré les fonctions pour lister les citoyens et les lieux de fêtes, permettant de visualiser que les chaines et les fonctions d'intégration étaient fonctionnelles. Ces fonctions se révéleront également utiles pour tester les fonctions d'ajout et de manipulation des listes du programme.

Pour l'ajout de citoyens et de fêtes, Il a suffi de créer une interface utilisateur légère demandant saisie des différentes données nécessaires à l'ajout d'un élément (nom, prénom...) puis d'appeler la fonction d'ajout manuel correspondant créé en amont. Il y a une petite variation pour l'ajout de lieu

qui nécessite la création d'une liste de participants et de ses bidons qui lui soit propre. Cette création a après coup été ajoutée dans la fonction d'ajout manuel de lieux.

L'ajout de participants à une fête s'est révélé plus compliqué. Il a fallu dans un premier temps créer une interface utilisateur demandant la date, a fête et le nom du participant. La fonction a d'ailleurs été faite de sorte a ce que sans pression de 0 a la fin du premier ajout, elle se lance une nouvelle fois pour permettre des ajouts multiples plus ergonomiques. Une fois le nom de la fête et du participant entrées, la fonction effectue une recherche parmi les listes « lieux » et « citoyen » pour retrouver les éléments correspondants afin de pouvoir pointer sur la liste de participants correspondant au lieu en question et ajouter le participant.

J'ai considéré dans cette partie qu'une fête correspondait à un lieu et donc que chaque fête avait une date unique appliquée à tous les citoyens.

Dans un premier temps la date était contenue dans un integer mais cela posait des problèmes de dépassement de mémoire, alors une structure DATE est désormais utilisée composée de trois integer « an », « jour » et « mois ».

Pour retrouver qui a été en contact avec un citoyen particulier, on cherche dans toutes les listes de participants des fêtes. Lorsque l'on relève une fois le nom de ce citoyen, on reprend la base de cette liste et on affiche tous ses participants.

L'interface a été pensée de sorte a ce que tous les contacts soient indiqués avec leur lieux dans un tableau lisible. On peut donc visualiser ou les contacts ont eu lieu et, si ils sont multiples avec la même personne.

L'enregistrement des fichiers situations se fait dans trois fichiers, « fetes.txt » « citoyens.txt » et « participants.txt ».

On lit les différentes listes et on écrit leurs données dans le fichier correspondant.

Le fichier « citoyens.txt » suit la syntaxe suivante :

Nom

Prenom

Nom

Prenom

...

Le fichier « fetes.txt » suit la syntaxe suivante :

Lieu

Lieu

Lieu

...

Le fichier « participants.txt » suit la syntaxe suivante :

Date.an

Date.mois

Date.jour

Nom

Prenom

Nom

Prenom

===== // séparateur

Date.an

Date.mois

Date.jour

Nom

Prenom

Nom

Prenom

=====

...

Pour ce qui est de la lecture des fichiers situations, on lit les fichiers en insérant les éléments à l'aide des fonctions d'ajout manuel créées initialement.

La suppression d'éléments antérieurs à une date fonctionne sur un principe simple : on navigue dans les différentes fêtes en extrayant la date de la fête d'un élément participant, après avoir demandé à l'utilisateur de saisir une date.

Nous vérifions par la suite pour chaque fête si l'année de cette date est inférieure à l'année saisie, le cas échéant, si le mois de cette date est inférieur à celui saisi et que l'année est la même et enfin dans le cas où le mois et l'année seraient identiques on vérifie si le jour est inférieur au jour saisi.

On fait pointer l'élément précédent sur l'élément suivant de la fête supprimée et vice versa, la fête n'apparaît alors plus dans la liste.

III- Démarche de test du programme

Tout au long du développement de nombreux tests ont été effectués au sein du programme.

Différents cas possibles :

- Test suite à un crash, j'ai alors utilisé des printf pour déterminer des nœuds où le programme bloquait et les instructions précises qui le faisaient crasher
- Test suite à un dysfonctionnement d'une fonction, j'ai alors utilisé des printf pour visualiser des variables clés et déterminer l'origine du problème.
- Test suite à l'ajout d'une fonction, j'ai alors testé le bon fonctionnement en simulant une utilisation du logiciel, ajout d'un citoyen, d'une fête...

IV – Perspectives d'amélioration

Pour améliorer le logiciel, il serait envisageable d'y ajouter des fonctions, ou encore de permettre une sauvegarde en ligne des situations, permettant l'utilisation d'une base de données commune entre plusieurs utilisateurs ou machines.

Il est également envisageable de permettre de créer différentes fêtes à différentes dates dans un même lieu en dissociant « lieux » et « fêtes ».

Pour conclure, ce projet m'a permis de suivre du début à la fin une démarche de construction d'un programme relativement important et ambitieux. Il m'a permis de me heurter à différents problèmes et à développer des techniques de débogage et de tests. Je suis plutôt satisfait du résultat qui, selon moi, répond au cahier des charges imposé. Je suis également assez satisfait de l'interface utilisateur que j'ai cherché à rendre le plus accessible possible.