



PROJET

Programmation en C

Réservation de tickets (Utilisateur)

groupe 2

Année académique 2016-2017

3^e Bachelier en Sciences Industrielles

Composition de l'équipe : 3TBSIEB

DETILLEUX Cyril
SULMON Pierre

Responsable de projet : M. JOJCZYK



VS



Table des matières

Table des matières.....	1
1 Introduction.....	2
1.1 Description du programme de base.....	2
2 Point à améliorer.....	3
2.1 Ajout des prix.....	3
2.2 Sauvegarder les réservations dans un second fichier.....	3
2.3 Vérifier la mise à jour du nombre de places dans le programme	3
2.4 Choix gestionnaire/utilisateur	3
2.5 Gestion des erreurs.....	4
3 Présentation du programme.....	4
3.1 Point de vue utilisateur.....	4
3.2 Point de vue du gestionnaire	6
4 Données manipulées	6
5 Fonctions importantes du programme	7
5.1 void initensembledestribune(struct ensembleDeTribune *cL).....	7
5.2 void initT (struct Tribune *T)	7
5.3 void telecharger_Tribunes(struct ensembleDeTribune *Ticket)	7
5.4 void telecharger_reservations(struct Tribune *T).....	7
5.5 void addTribune(struct ensembleDeTribune *cL, struct Tribune newT).....	8
5.6 void addPersonne (struct Tribune *T, struct Personne newP).....	8
5.7 void PROGLOOPTribuneEnplus(struct ensembleDeTribune *cL).....	8
5.8 void saveTRIBUNE(struct ensembleDeTribune *cL)	9
5.9 void save_reservations	9
6 Conclusion.....	10
6.1 Points à améliorer	10

1 Introduction

Dans le cadre du cours d'informatique de 3BSI, nous avons été amenés à développer un programme de réservation de tickets d'un point de vue utilisateur. Celui-ci sera codé en langage C. Nous utilisons comme base le programme des années précédentes, le but étant qu'au fil des années les projets se perfectionnent et deviennent de plus en plus étoffés.

Notre programme permet de réserver des tickets pour la finale de NBA opposant les Cavaliers de Cleveland aux Warriors de Golden State.

Nous partons d'une base de données préétablie par nos soins sur un fichier texte, le programme se servira de ce fichier « Tribune.DATA » afin de récupérer les informations quant aux tribunes.

1.1 Description du programme de base

Nous nous basons donc sur un projet existant, le premier travail à effectuer est donc de comprendre le programme de base ainsi que l'entièreté de ses fonctions. Notre but étant d'apporter des modifications sur certains aspects.

Lorsqu'on lance le programme, nous avons donc 4 choix :

- 1) Ajouter des tribunes
- 2) Afficher toutes les tribunes
- 3) Sauvegarder les tribunes dans un fichier
- 4) Réserver un ticket

La première des fonctions sert à ajouter des tribunes dans la console. Dans la deuxième fonction le programme affichera les différentes tribunes qu'il a été lire dans un fichier texte « Tribune.DATA », mais également les tribunes ajoutées. On pourra ensuite sauvegarder le tout dans ce fichier texte.

La dernière fonction est celle qui nous intéresse particulièrement, car elle est liée à l'utilisateur et permet donc à celui-ci de choisir une tribune et enfin de rentrer ces données personnelles.

Pour plus d'informations concernant le code, voir rapport de base « Rapport-projet-info-AKAN-DI LEGAMI ».

2 Point à améliorer

2.1 Ajout des prix

Premièrement le programme nécessite quelques précisions quant à la réservation des tickets, nous allons donc ajouter un critère prix pour chaque tribune du stade. Nous aurons 3 formules de prix différentes suivant les tribunes :

- La tribune économique à 35 \$
- La tribune de base ou regular à 45\$
- La tribune VIP à 70\$

Ce critère a été ajouté dans le fichier « Tribune.DATA », et dans le programme.

2.2 Sauvegarder les réservations dans un second fichier

Une fonction essentielle pour nous dans la réservation de tickets est donc de pouvoir lister toutes les réservations effectuées pour l'événement.

On créera donc un fichier « DONNEES_UTILISATEURS.txt », le programme va y écrire l'ensemble des réservations avec les données personnelles de chacune des réservations ; Nom, Prénom, Téléphone, Date de naissance, Numero de tribune et nombre de tickets réservés.

2.3 Vérifier la mise à jour du nombre de places dans le programme

L'utilisateur désire un nombre bien défini de tickets dans la tribune de son choix. Les tribunes possèdent un nombre limité de places (défini dans le fichier « Tribune.DATA »), il faut donc que lorsque l'utilisateur réserve « X » places dans une tribune « Y », ce nombre de places soit disponible. Une condition devra donc être remplie pour pouvoir réserver ce nombre de places. De plus, après réservation du nombre de places, il faudra décompter le nombre « X » de places réservées par l'utilisateur dans le fichier texte et donc se mettre à jour à chaque fin de réservation.

2.4 Choix gestionnaire/utilisateur

Le programme de base ne répond pas vraiment au critère demandé à savoir : se placer d'un point de vue utilisateur.

En effet pour le moment l'utilisateur peut ajouter des tribunes en plus et les sauvegarder dans le fichier, ce qui est une fonctionnalité propre au gestionnaire. Nous allons donc séparer dès le départ le programme existant en deux, une part pour le gestionnaire où l'on placera les fonctions existantes quant à l'ajout des tribunes.

Et d'autre part une partie destinée à l'utilisateur, dans laquelle il pourra voir l'ensemble des tribunes de l'événement, choisir l'équipe qu'il supporte et finaliser son choix en rentrant ses données personnelles. C'est dans cette seconde partie que nous allons opérer les différentes améliorations.

2.5 Gestion des erreurs

Enfin, on tentera de corriger les erreurs dues à l'utilisateur. En effet, on s'est basé jusqu'ici sur l'hypothèse que l'utilisateur rentre des données correctes à chaque instruction, mais ce n'est pas toujours le cas.

Il peut commettre des erreurs sur les choix disponibles mais également sur les informations qu'il rentre.

Nous avons donc créé une restriction sur les données rentrées par l'utilisateur :

- Sur les choix qu'il a, si cela ne correspond pas à une valeur, on lui demandera s'il veut recommencer l'opération
- Sur sa date de naissance, qui devra être de la forme « DD/MM/YYYY ».

3 Présentation du programme

```
-----MENU PRINCIPAL-----
Etes vous utilisateur ou gestionnaire?
1. Gestionnaire
2. Utilisateur
```

Nous avons donc commencé par diviser le programme en 2, une partie consacrée à l'utilisateur et l'autre pour le gestionnaire.

On demande donc le choix à l'utilisateur du programme.

3.1 Point de vue utilisateur

```
Bienvenue au match final de la NBA!

------(1)-----

      [---]
      [ ]
      [MM]
      |
      |
      |
    // | \\
    // | \\
(2) // | \\ (3)
    // | \\
    // | \\
    // | \\
(4) // | \\ (5)
    // | \\
    // | \\

      |
      |
      |
      [---]
      [MM]
      [ ]
      [---]

------(6)-----

        Quelle equipe supportez-vous?

1.Cavaliers de Cleveland
2.Warriors de Golden State
(Tapez 1 ou 2)
Votre choix:
```

On entre donc l'interface destiné à l'utilisateur.

Un peu de fun ! On a réalisé une visualisation du terrain de basket afin que l'utilisateur connaisse l'emplacement des tribunes pour pouvoir par la suite décider dans laquelle il souhaite se placer.

On demande donc quelle équipe supporte l'utilisateur.

```

-**-*-Supporter des Cavaliers-**-*-
Tribune des CAVALIERS (1)eco
-----
Places restantes: 970
Prix de la place: 35 $

Tribune des CAVALIERS (2)regular
-----
Places restantes: 977
Prix de la place: 45 $

Tribune des CAVALIERS (3)VIP
-----
Places restantes: 974
Prix de la place: 70 $

Vous avez le choix parmi les tribunes allant de 1 a 3
Quelle tribune choisissez-vous? _

```

On a donc la liste des différentes tribunes et l'utilisateur peut faire son choix parmi les tribunes correspondant à l'équipe qu'il supporte.

```

Personnes inscrites
-----
Nom: Detilleux
Prenom: Cyril
Numero telephone: 0493497770
Date de naissance: 03/05/1996
Tribune 2
Places reservees: 3

Place restante dans la tribune: 965

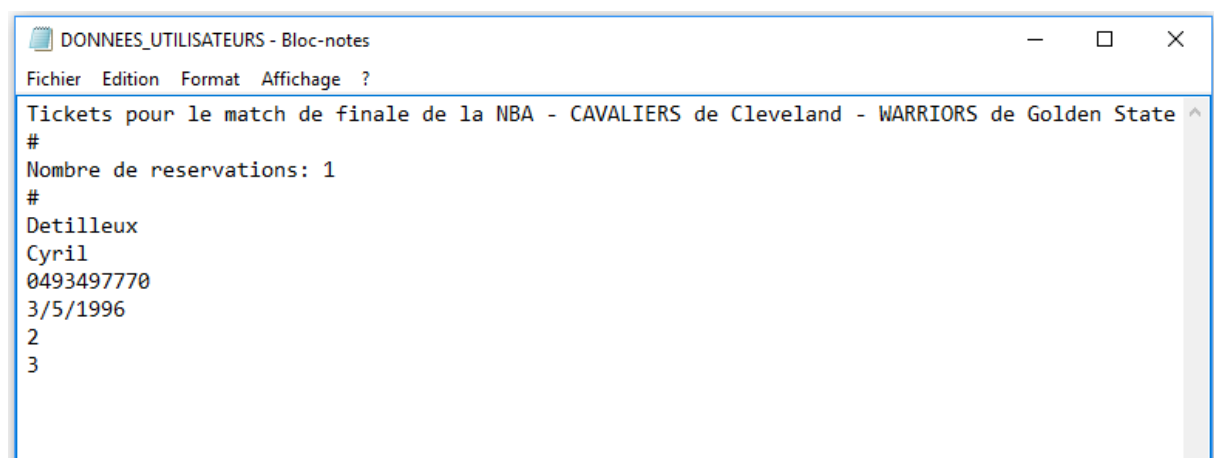
Saving Contact List
-----
Done!

Voulez-vous revenir au menu principal? (oui/non)

```

L'utilisateur rentre ensuite ses informations personnelles. Le programme liste ensuite les informations des personnes inscrites et le nombre de places restantes dans la tribune où les places ont été réservées.

On sauvegarde ces informations dans un fichier txt « DONNEES_UTILISATEURS.txt »



3.2 Point de vue du gestionnaire

L'interface gestionnaire reste presque identique à celle du programme de base. La différence étant que si l'on choisit l'option 4 on sort du menu gestionnaire et on revient au menu principal. En effet il est maintenant inutile, en tant que gestionnaire, de passer dans la partie utilisateur après avoir géré les tribunes du stade.

```
1 - TRIBUNE EN PLUS
2 - AFFICHER TOUTES LES TRIBUNES
3 - SAUVEGARDER LES TRIBUNES DANS LE FICHIER
4 - TERMINER
```

4 Données manipulées

Le programme contient une structure principale appelée ici « ensembleDeTribune » qui va reprendre l'ensemble des tribunes du stade. Elle contient une chaîne de caractère reprenant le nom de cet ensemble.

Une deuxième structure appelée ici Tribune contient 3 champs :

- Entier pour le nombre de places
- Entier pour le prix de la place
- Une chaîne de caractère pour le nom de la tribune.

Une troisième structure appelée Personne contient 5 champs :

- Une chaîne de caractère pour le nom
- Une chaîne de caractère pour le prénom
- Une chaîne de caractère pour le numéro de téléphone
- Un entier pour le nombre de tickets réservés
- Un entier pour la tribune dans laquelle ces derniers sont réservés.

Une quatrième structure appelée Date contient 3 champs :

- Un entier pour le jour
- Un entier pour le mois
- Un entier pour l'année

La particularité de ces structures est que chacune est reprise par une autre. En effet, la structure ensembleDeTribune possède un pointeur qui pointe vers la structure Tribune. Cette dernière comprend elle-même la structure Personne qui va reprendre la structure Date.

5 Fonctions importantes du programme

Tout d'abord pour faciliter la compréhension du programme dont nous avons hérité, nous avons commencé par scinder le projet en 3 parties : le main, les fonctions et les structures avec la déclaration des fonctions. Ces 3 parties sont reprises dans le projet par les fichiers « main.c », « Functions.c » et « Structures.h ».

5.1 void initensembledestribune(struct ensembleDeTribune *cL)

Cette fonction va permettre d'initialiser l'ensemble des tribunes à zéro. Pour se faire on va utiliser un pointeur qui va venir pointer chacune des tribunes via une boucle « for » et va initialiser le nom de chaque tribune à une chaîne de caractère vide. Cela se fait en venant y insérant le caractère \0.

5.2 void initT (struct Tribune *T)

Par la suite on viendra également initialiser les personnes que contient chaque tribune à zéro de la même manière que pour l'initialisation des tribunes. À nouveau cela se fait en attribuant à la chaîne de caractère « nom » de la personne le caractère \0.

5.3 void telecharger_Tribunes(struct ensembleDeTribune *Ticket)

Cette fonction permet de chercher les tribunes dans un fichier qui se trouve dans l'exécutable. Celui a été nommé «Tribune.DATA». La fonction fopen permet d'ouvrir le fichier, il prend en argument le nom du fichier et le mode d'utilisation. Dans notre cas on utilise le fichier en mode lecture seule, "r". Cela nous permet ainsi de récupérer les informations concernant les tribunes dans le fichier.

Après avoir vérifié que nous sommes bien dans ce dernier, une boucle va venir récupérer le nom, le nombre de places disponibles et le prix de la place de chacune des tribunes. La boucle ne sera pas quittée tant que tous les éléments ne seront pas récupérés. La variable list_size permet de stocker le nombre de tribunes présentes dans le fichier et ainsi de toutes les télécharger.

Celles-ci sont stockées dans une structure déclarée dans la fonction et nommée « newT ». Le programme fera ensuite appelle à la fonction « addTribune » pour attribuer chaque tribune récupérée à l'ensemble de toutes les tribunes du stade.

5.4 void telecharger_reservations(struct Tribune *T)

Cette fonction est semblable à la précédente à la différence qu'elle va chercher toutes les réservations de clients dans la structure Tribune. Elle va lire les informations dans le fichier nommé « DONNEES_UTILISATEURS.txt ».

5.5 void addTribune(struct ensembleDeTribune *cL, struct Tribune newT)

L'utilité de cette fonction vient d'être expliquée ci-dessus. Elle prend 2 paramètres en compte : la structure ensembleDeTribune qui contient notre liste de tribune et une structure Tribune appelée newT qui contient les données de la tribune que l'on va ajouter dans le stade.

L'ensemble des « if », « else » représente les cas suivants :

- Liste pleine : affichage d'une erreur
- Liste vide : ajout de la Tribune en position 1.
- Liste avec des tribunes déjà présentes : on place la nouvelle tribune au bon endroit. Cela se base sur le résultat de la fonction strcmp. Cette fonction permet de placer les noms dans l'ordre alphabétique. La fonction strcmp étant utilisée ici, on devra ajouter la librairie string.h en haut du programme.

5.6 void addPersonne (struct Tribune *T, struct Personne newP)

Le fonctionnement est exactement le même que la fonction « addTribune » sauf que les paramètres sont ici la structure Tribune et la structure Personne nommée newP. Cette dernière contient les informations de l'utilisateur qui a réservé ses billets pour le match et qu'on va donc placer dans la tribune qu'il a choisie.

5.7 void PROGLOOOPtribuneEnplus(struct ensembleDeTribune *cL)

On rencontre cette fonction lorsque l'on choisit l'option gestionnaire dans le menu principal.

Elle présente le menu suivant qui fonctionne via un switch:

```
1 - TRIBUNE EN PLUS
2 - AFFICHER TOUTES LES TRIBUNES
3 - SAUVEGARDER LES TRIBUNES DANS LE FICHIER
4 - TERMINER
```

- Case 1 : On désire ajouter une tribune au stade. Pour cela on entre les informations relatives à la nouvelle tribune que l'on désire créer. On fait ensuite appelle à la fonction « addTribune » décrite précédemment pour l'ajouter dans l'ensemble des tribunes.
- Case 2 : On affiche l'ensembles des tribunes via une simple fonction « printTribune » dans laquelle on retrouve une simple boucle « for » qui permet de lister chaque tribune avec ses informations.
- Case 3 : On utilise la fonction « saveTRIBUNE » pour enregistrer la tribune que l'on a ajoutée au case 1. Il est ensuite nécessaire de réinitialiser les tribunes à zéro et de retélécharger le fichier qui vient

d'être modifié pour que cette nouvelle tribune soit prise en compte pour la suite de l'exécution du programme.

```
case 3:
    saveTRIBUNE (cL);
    initensembledestribune (cL);
    telecharger_Tribunes (cL);
    break;
```

- Case 4 : Une fois avoir fini la gestion du stade et de ses tribunes, on peut sortir du mode gestionnaire et revenir au menu principal

Remarque : Dans ce menu gestionnaire, si l'on souhaite rajouter des tribunes on est obligé d'en rajouter un nombre pair. En effet un stade comporte toujours un nombre pair de tribunes.

5.8 void saveTRIBUNE(struct ensembleDeTribune *cL)

Cette fonction va permettre d'aller inscrire les informations relatives aux tribunes dans un fichier nommé « TRIBUNE.DATA ». Pour se faire, on utilise la fonction « fopen » en mode écriture (w) et on vérifie que l'on ouvre correctement le fichier.

- On commence par y inscrire quelques notations permettant de structurer le fichier ainsi que le nombre de tribunes présentes dans le stade en utilisant la fonction « getListSize_TRIBUNES » : celle-ci est composée simplement d'une boucle « for » qui incrémente un compteur en décomptant le nombre de tribunes présentes dans la structure « ensembledeTribune ».
- On fait ensuite appelle à la fonction « recordTribune » qui présente une boucle « while » permettant d'inscrire les informations de chaque tribune l'une à la suite de l'autre, et ce jusqu'à ce qu'elles aient toutes été passées en revue. Chaque tribune est séparée d'une autre par un « # ». Dans le fichier cela se présente comme suit :

#	→ Informe qu'on commence une nouvelle tribune
Tribune des CAVALIERS (1)eco	→ Nom de la tribune
989	→ Nombre de places restantes
35	→ Prix de la place
#	→ Informe qu'on en a fini avec cette tribune

5.9 void save_reservations

On va enregistrer les informations des clients le fichier de l'exécutable nommé « DONNEES_UTILISATEURS.txt ». On utilise cette fois la fonction « getListSize_RESERVATIONS » pour recenser le nombre de réservations total ayant été effectué et ensuite la fonction « record_Reservation » qui contient la boucle while permettant d'inscrire les informations de chaque personne.

6 Conclusion

Pour conclure, nous pouvons affirmer que nous avons atteint les objectifs établis avec M. JOJCZYK, qui étaient d'ajouter les prix, d'enregistrer les données dans un fichier et que les places disponibles dans les tribunes se mettent à jour automatiquement. Nous avons également séparé le programme en 2 interfaces : utilisateur et gestionnaire.

6.1 Points à améliorer

Après c'est quelques améliorations, notre programme n'est pas parfait et dispose de voies d'améliorations afin de perfectionner celui-ci :

- Lors de l'encodage d'une nouvelle tribune, il y a un problème lorsqu'on nomme la nouvelle tribune. En effet les tribunes se répartissent dans le fichier dans l'ordre alphabétique ce qui fait qu'elles prendront une position bien spécifique dans le fichier. C'est un problème car si par exemple on rajoute 2 tribunes, 1 pour les Cavaliers et une pour les Warriors, et que de par leur dénomination elles se placent en dernière position dans le fichier, alors lorsque l'utilisateur va choisir son équipe il n'aura pas les bonnes tribunes réservées à celle-ci. Il faudrait donc pour cela ajouter d'office une tribune aux Cavaliers et une aux Warriors et incrémenter le nombre de tribunes de chacune des équipes de 1.

```
Quelle equipe supportez-vous?
1.Cavaliers de Cleveland
2.Warriors de Golden State
(Tapez 1 ou 2)
1
--*-*-Supporter des Cavaliers--*-*-
Tribune des CAVALIERS (1)eco
-----
Places restantes: 961
Prix de la place: 35 $

Tribune des CAVALIERS (2)regular
-----
Places restantes: 964
Prix de la place: 45 $

Tribune des CAVALIERS (3)VIP
-----
Places restantes: 984
Prix de la place: 70 $

Tribune des WARRIORS (4)eco
-----
Places restantes: 1996
Prix de la place: 35 $

Vous avez le choix parmi les tribunes allant de 1 a 4
Quelle tribune choisissez-vous?
```

- Une fonction utile serait de calculer le total à payer pour chaque réservation dans le fichier "DONNEES_UTILISATEURS.txt".
- Au début du programme, lorsqu'on demande à la personne de choisir s'il est utilisateur ou gestionnaire, le gestionnaire a accès à des modifications de tribunes. Cette partie est donc strictement réservée au gestionnaire et il serait bon de la protéger avec un mot de passe.
- Dans la fonction PROGLOOOPtribuneEnplus de la partie gestionnaire, il serait intéressant d'intégrer l'option 3, qui est d'enregistrer les modifications des tribunes, à l'option 1 qui permet d'enregistrer de nouvelles tribunes.

Ce projet nous a donc permis d'apprendre l'utilisation de chaîne de caractère, d'écriture et de lecture dans des fichiers texte et d'en apprendre plus sur le langage C. La programmation s'apprend en pratiquant et ce projet nous a permis d'évoluer dans le domaine de la programmation.