



Rapport Projet Informatique - CY'FEST

Informatique 2 - M. Romuald Grignon

Jad Messous
Ashana Maheswaran

Mai 2024

Table des matières

Table des matières.....	2
Description de l'équipe et du sujet.....	3
Organisation de l'équipe.....	4
Problèmes rencontrés et les solutions apportées.....	5
Résultats.....	7

Description de l'équipe et du sujet

Notre équipe est constituée de deux personnes : Jad Messous et Ashana Maheswaran. Nous sommes le groupe F dans la classe MI 1.

Nous avons choisi le sujet CY'FEST qui est un projet de type gestion. En effet, le projet consiste à gérer des scènes de festival.

Notre programme peut être utilisé par les personnes souhaitant réserver une place à un concert mais également aux managers du festival. Ainsi, le programme présente plusieurs fonctionnalités permettant de gérer un festival.

Les festivaliers peuvent : voir les salles des concerts disponibles, réserver des sièges.

Les managers peuvent : créer des salles, attribuer un concert à une salle, définir l'heure de fin du concert, modifier les salles et observer leur état.

Le but est de permettre à des participants de réserver leurs places de festivals ainsi qu'au manager des salles à gérer, créer et modifier ces salles.

Ce projet nécessite ainsi la maîtrise des fichiers, de l'allocation dynamique ainsi que de l'affichage.

Organisation de l'équipe

Tout d'abord, notre première tâche était de bien comprendre les fonctionnalités du projet et d'identifier les différentes fonctions que l'on devait coder. Ainsi, nous avons décidé de faire un arbre avec les différentes procédures à programmer afin de bien comprendre le fonctionnement et les liaisons entre les fonctions.

Ensuite, avant la première séance de TD, nous avons décidé de coder chacun de notre côté quelques fonctions, notamment le "menu principal" du projet qui permet à l'utilisateur de choisir entre le menu manager et festivalier.

Lors de notre première séance de TD, nous avons créé un lien sur le site *codeshare.io*, qui nous permettait de mettre notre code en commun et pouvoir le modifier chacun de notre côté. Cela a été très utile car nous avions chacun accès au code. Nous avons ainsi pu discuter des fonctions qu'on a codé et nous nous sommes mis d'accord pour la version finale.

Puis, à l'aide de l'arbre que l'on avait fait au début et du *codeshare*, on répartissait les fonctions à coder et on vérifiait son fonctionnement ensemble lors des séances de TD.

Puis quelques jours avant le rendu du projet, on a fait des appels afin d'identifier ce qu'on a fait et ce qui nous restait à faire et voir ce qui était prioritaire : nous nous sommes également rendu à la bibliothèque afin de faire les dernières modifications du programme ensemble et voir les améliorations qu'on pouvait ajouter.

Enfin, nous avons fait le README et le Makefile et tout déposé sur GitHub.

Problèmes rencontrés et les solutions apportées

Nous avons rencontré de nombreux problèmes au cours de notre projet. Voici, les différents problèmes rencontrés et les solutions qu'on a apportées.

- 1) Dans notre programme, nous avons besoin à quelques endroits d'afficher les différentes salles créées et nous voulions créer une variable globale, mais nous n'avions pas le droit. Or, par manque de temps, nous avons décidé de garder la variable globale car c'était le plus simple pour nous.
- 2) Durant notre projet, nous avons voulu utiliser la bibliothèque `"#include <dirent.h>` mais nous avons décidé de ne pas le faire car c' était compliqué pour nous et hors programme.
- 3) Lorsque qu'on programmait tout ce qui était attribution d'un concert dans une salle, nous avions du mal à accéder aux données du concert et de les sauvegarder, ainsi nous avons créé une structure "Concert", qui est constitué d'une salle, du nom de l'artiste ainsi de la structure "struct tm" pour le temps.
- 4) Il y a quelques notions que nous n'avions pas bien compris comme les fichiers alors nous avons regardé des vidéos YouTube qui nous ont aidé à mieux comprendre
- 5) Lorsqu'on bloquait sur des bouts de fonctions, comme par exemple la sauvegarde des modifications dans les fichiers, on s'aidait avec ChatGPT pour voir à quoi ressemblent les fonctions et mieux comprendre leur fonctionnement.
- 6) Nous avons également rencontré un problème lors de la création de salles. En effet, lorsque l'utilisateur veut accéder à ses données (afficher le salle par exemple), le programme demande de taper `"/nomdelasalle.txt"` au lieu du nom de salle toute seule
- 7) Puis, nous ne savions pas bien comment moduler le programme et par manque de temps, nous avons décidé de ne pas le faire.

Un des problèmes majeurs était la robustesse du programme. En effet, comme il y a beaucoup de saisie d'utilisateur, il fallait que le programme soit robuste.

- 8) Lorsqu'on demande au manager d'entrer un nom de salle, il faut qu'il entre un nom qui ne dépasse pas la taille du tableau qu'on a codé. Pour cela, nous avons écrit `"%50s"`, pour que le programme ne garde que les 50 premiers caractères (taille de notre tableau), au lieu de redemander au manager de taper un nouveau nom.

- 9) Pour les sécuriser les scanf, nous avons créé une fonction pour vider le buffer en cas d'entrée invalide lors du scanf
- 10) Il y a beaucoup de saisies d'utilisateurs, donc il y a beaucoup d'endroits où l'on doit s'assurer que l'utilisateur ne puisse pas mettre des valeurs invalides. Mais, comme nous sommes seulement deux personnes dans ce projet au lieu de trois, la charge de travail était plus importante et nous n'avions pas de rendre le programme le plus robuste possible. Ainsi, il y a des endroits dans notre code qui vont mal fonctionner lors d'une utilisation anormale.

Résultats

Ainsi, notre programme fonctionne de manière normale mais n'est pas aussi robuste que demandé par manque de temps et de personnes. De plus, nous n'avons pas le temps d'ajouter des améliorations. Mais, nous avons eu l'idée de créer un mot de passe pour accéder au mode manager, or, par manque de temps, nous ne l'avons pas fait.