

# CY FEST

## programmation de salle de concert en langage C



### Introduction :

Nous avons choisi le sujet CY FEST car nous sommes passionnés par la musique et les concerts. Ensuite, le thème de la gestion de la réservation de places est en lien direct avec notre cursus en double diplôme de management. Ce sujet nous permet de combiner notre intérêt personnel pour les événements musicaux avec les compétences professionnelles que nous développons dans le cadre de notre formation, manager de la transformation numérique.

### Rôles de chacun :

#### - Corentin :

A modélisé le programme en pseudo-code, modélisation schématique

A codé la structure générale du programme, celui qui maîtrise le mieux le langage C d'entre nous trois et vérifie les erreurs des autres.

A corrigé la plupart des erreurs de programmation et développer les fonctionnalités supplémentaires

**- Hugo :**

A réalisé une partie de la fonction création salle (assistance partielle en programmation)

A réparti les tâches de chacun

A rédigé le rapport CY FEST

**- Ilias :**

A rédigé le README

Création espace Github

A réalisé une partie du programme (assistance partielle en programmation)

Pour avancer de manière efficace sur ce projet, il était essentiel de définir clairement les rôles de chacun. Plutôt que de nommer un chef de projet unique, nous avons opté pour une répartition des responsabilités basée sur les compétences individuelles de chacun. Cette approche a permis à chaque membre de devenir autonome et de progresser dans la réalisation du programme en travaillant sur les différents fichiers créés. Tout au long du projet, notre groupe a démontré une dynamique collaborative et une progression constante.

## Comportement du programme :

*Il contient deux modes :*

### **Mode Manager :**

- Créer une ou plusieurs salles de concerts
- Attribuer un nom aux salles
- Possibilité de choisir le nombre de rangées de catégorie A et B le nombre de sièges par rangée et le nombre de sièges par catégories.
- Calcul différence pour catégorie C
- Supprimer un concert lorsque ce dernier est terminé et changer le nombre de catégories A et B
- Condition Booléenne Fosse si oui, la catégorie A devient une fosse
- Attribution du prix d'un emplacement de chaque catégorie
- Définir l'heure d'un concert

### **Mode Festivalier :**

- Accès aux salles où les concerts ne sont pas terminés.
- Peut réserver des places dans ces salles.
- Peut choisir la rangée et le siège qu'il souhaite réserver, ou opter pour une place dans la fosse si la salle en dispose.

## Problèmes rencontrés :

- Au début du projet, nous avons rencontré des difficultés pour répartir le travail de manière équitable. La diversité des compétences et des expériences au sein du groupe a rendu cette tâche complexe. Cependant, nous avons réussi à surmonter cette première difficulté en prenant en compte les forces et les préférences de chacun.
- La modélisation du programme en pseudo-code s'est avérée difficile, en particulier en suivant les consignes du PDF distribué. Par exemple, la fonction `methodeCreationSalle`, qui inclut la répartition des rangées et des sièges par rangée, a nécessité plusieurs itérations pour être correctement structurée.
- Nous avons également rencontré des problèmes de compatibilité entre différents systèmes d'exploitation et éditeurs de texte, car nous utilisons trois machines différentes :
- *PC HP sous Debian* : Nous avons rencontré des problèmes mineurs liés à des paquets non mis à jour et à des restrictions utilisateur, qui ont été résolus en déverrouillant les restrictions et en redémarrant la machine.
- *ASUS Chromebook sous ChromeOS* : Cet ordinateur, utilisé principalement pour le codage, n'a rencontré aucun problème majeur et fonctionnait parfaitement avec Visual Studio Code.
- *Macbook Air M3 sous MacOS* : Nous avons résolu les problèmes initiaux en installant des extensions spécifiques pour MacOS. Malgré quelques messages d'erreur persistants dans le terminal d'exécution, le programme s'exécutait correctement.
- Nous avons rencontré des problèmes d'exécution dus à des erreurs de segmentation et à des oublis de ponctuation, comme des points-virgules à la fin des instructions ou des accolades manquantes.
- La suppression de fichiers posait problème : bien que le programme indiquait qu'un fichier avait été supprimé, celui-ci restait visible dans l'explorateur de fichiers.

- Lors de la suppression de dossiers, des caractères indésirables se glissaient dans les commandes de saisie, particulièrement suite à la saisie d'un `scanf` et sur les fichiers `.txt`.
- Nous avons dû gérer la création de fichiers et les enregistrer dans dossier spécifique, cela ne fonctionnait pas au début.
- La sélection des salles par leur numéro fonctionnait, mais l'option de continuer à utiliser une salle posait des problèmes de variable. Nous avons donc transformé des lettres en nombres pour les utiliser correctement.

### **Fonctionnalité supplémentaires :**

- Mot de passe implémenté
- Création nom d'utilisateur
- Scène numéroté
- Couleur et affichage embellie

## Annexe du CY FEST :

Cy Fest

Salle = "nom" + "nom.txt"  
 = [Salle A] + [B] + [C]  
 = :Occupe: + :Fosse:  
 Concert = "nom" + "Artiste" + hddat + hfin +  
 prix A + prix B + prix C

### Menu

#### Mode manager

- ~~Attribution concert~~
- ~~Attribution concert~~
- Attribution concert  
(Fosse, Prix A, Prix B, Prix C, hfin)
- Changement Config  
(Fosse, Prix A, Prix B, Prix C)
- Vo des Salles  
↳ Nom, Occupe, Sieges, Prix

#### Mode Festivalier

- Vo des concert  
↳ Nom, Artiste, Place
- Reservation  
↳ Place

1. Mode Festivalier
2. Mode Manager
3. Quitter

→

### Menu

1. Renvoie vers Mode Festivalier
2. Renvoie vers la page Mode passe
3. Sauvegarde et quitte

Nom :  
 Fosse :  
 Place A :  
 place B :  
 place C :  
 Valider :

### Changement config

Si autre → Erreur → Mode Manager

### Salle 1 :

Statut : Libre / Occupe  
 Fosse : Avec / Sans  
 nb place total : 30 / 400  
 nb place A / Fosse : 30 / 100  
 nb place B : 20 / 200  
 nb place C : 40 / 100  
 recette total : 180€

→

### Infos Salle

Si 0 → Mode Manager  
 Si 1 → Changement Config

Salle 1 : Concert by Artiste

Scene

R <sub>1</sub>	0	x	x	0	x	0	x	x
R <sub>2</sub>	0	x	x	0	x	0	0	0
R <sub>3</sub>	0	0	0	0	x	0	0	0
R <sub>4</sub>	0	0	0	x	0	x	0	0
R <sub>5</sub>	x	0	x	x	x	0	x	0
R <sub>6</sub>	0	0	0	0	0	x	0	0
R <sub>7</sub>								
	C <sub>1</sub>	C <sub>2</sub>	C <sub>3</sub>	C <sub>4</sub>	C <sub>5</sub>	C <sub>6</sub>	C <sub>7</sub>	C <sub>8</sub>

Légende: .....

Pour réserver en ligne ranger/cabon

Vo Salle

Si 0 → Menu Festivalier

Si Rnb/Cnb Libre → Réserve (Recap) → Menu Festivalier

Si Rnb/Cnb Occupé → Arrêt refuser → Menu Festivalier

Si autres → Mode Festivalier

nom Salle : →

Avec fosse: →

nb total desiege: →

nb siege A: Si fosse Non →

nb siege B: →

Valider la salle → o/n → Mode Manager

Crée Salle

Occupé → Libre par default

Si autre → Erreur Mode Manager

Salle	statut	total	premier	Prix
1. Salle 2	Libre	400	0	0
2. Salle 3	Occupé	200	22	44 €

A. Crée une salle

B. Attribuer un concert

C. Menu

Tu vas faire quoi →

Mode Manager

A → crée une salle

B → Attribuer un concert

C → Menu

nb → info Salle

Autre → Erreur → Mode Manager

Liste - salle = [ Salle 1, Salle 2, Salle 3, ... ]

↓

structure de donnée

nom = "NULL"

1. Concert A by Artiste • Salle A • Complet
2. Concert B by Artiste • Salle B • Complet
3. Concert C by Artiste • Salle C • 14 places
4. Concert D by Artiste • Salle D • 3 places
0. Retour Menu

Pour réserver choisissez un concert : ↘

### Mode Festivalier

Si aucun nombre ↘ Non reconnu

Si 0 ↘ Menu

Si nb ↘ Vusalle

Nom utilisateur : ↘

Mot de Passe : ↘

Accès Refusé

↘  
Menu

Accès Accepté

↘  
Mode Manager

### Mot de Passe

Fichier Mdp avec les uti et mdp

Nom Concert :

Nom Artiste :

heure de début :

heure de fin :

Prix A ou fosse :

Prix B :

Prix C :

Salle existante libre :

Valider : yn ↘ Mode Manager

### Affectuer Concert

Si autres ↘ Erreur ↘ Mode Manager

Code Erreur ?

les 12 pages de schématique avant pseudocode et passage en C (ordre aléatoire)