



v1.0 **Projet** *CY'Fest*

CLASSE préING1 • 2023-2024

AUTEURS Eva ANSERMIN - Romuald GRIGNON

E-MAILS eva.ansermin@cyu.fr - romuald.grignon@cyu.fr

DESCRIPTION DU PROJET

Le but de ce projet est d'écrire un programme permettant à des organisateurs de festival de gérer les différentes scènes du festival CY'FEST.

Ce festival est un peu particulier car les participants doivent payer et réserver leurs places pour les différents concerts.

Ce projet est donc un projet de type gestion qui va notamment nécessiter une bonne maîtrise :

- des fichiers
- de l'allocation dynamique
- de l'affichage (interface)

INFORMATIONS GENERALES

Taille de l'équipe

Ce projet est un travail d'équipe. Il est autorisé de se réunir en groupe de 3 au maximum. Si le nombre d'étudiants n'est pas un multiple de 3 et/ou si des étudiants n'arrivent pas à constituer des groupes, c'est au chargé de TD de statuer sur la formation des groupes. Pensez-donc à anticiper la constitution de vos groupes pour éviter des décisions malheureuses.

Démarrage du projet

Vous obtiendrez de plus amples informations quant aux dates précises de rendu, de soutenance, les critères d'évaluation, le contenu du livrable, ..., quand le projet démarrera officiellement.

Dépôt de code

Vous devrez déposer la totalité des fichiers de votre projet sur un dépôt central Git. Il en existe plusieurs disponibles gratuitement sur des sites web comme github.com ou gitlab.com.

Rapport du projet

Un rapport écrit est requis, contenant une brève description de l'équipe et du sujet. Il décrira les différents problèmes rencontrés, les solutions apportées et les résultats. L'idée principale est de montrer comment l'équipe s'est organisée, et quel était le flux de travail appliqué pour atteindre les objectifs du cahier des charges. Le rapport du projet peut être rédigé en français.

		<p>Démonstration</p> <p>Le jour de la présentation de votre projet, votre code sera exécuté sur la machine de votre chargé(e) de TD. La version utilisée sera la dernière fournie sur le dépôt Git avant la date de rendu. Même si vous avez une nouvelle version qui corrige des erreurs ou implémente de nouvelles fonctionnalités le jour de la démonstration, c'est bien la version du rendu qui sera utilisée.</p> <p>Organisation</p> <p>Votre projet complet devrait (dans l'idéal) être stocké sur un dépôt git (ou un outil similaire) tout au long du projet pour au moins trois raisons : éviter de perdre du travail tout au long du développement de votre application, être capable de travailler en équipe efficacement, et partager vos progrès de développement facilement avec votre chargé(e) de projet. De plus il est recommandé de mettre en place un environnement de travail en équipe en utilisant divers outils pour cela (Slack, Trello, Discord, ...).</p>
	<p>CRITERES GENERAUX</p>	<ul style="list-style-type: none"> • Le but principal du projet est de fournir une application fonctionnelle pour l'utilisateur. Le programme doit correspondre à la description en début de document et implémenter toutes les fonctionnalités listées. • Votre code sera généreusement commenté. • Tous les éléments de votre code (variables, fonctions, commentaires) seront écrits dans la même langue. Langue anglaise conseillée mais pas obligatoire. • Votre application ne doit jamais s'interrompre de manière intempestive (crash), ou tourner en boucle indéfiniment, quelle que soit la raison. Toutes les erreurs doivent être gérées correctement. Il est préférable de d'avoir une application stable avec moins de fonctionnalités plutôt qu'une application contenant toutes les exigences du cahier des charges mais qui plante trop souvent. Une application qui se stoppe de manière imprévue à cause d'une erreur de segmentation ou d'une exception, par exemple, sera un événement très pénalisant. • Votre application devra être modulée afin de ne pas avoir l'ensemble du code dans un seul et même fichier par exemple. Apportez du soin à la conception de votre projet avant de vous lancer dans le code. • Le livrable fourni à votre chargé(e) de TD sera simplement l'URL de votre dépôt Git accessible publiquement. Même si vous n'avez pas utilisé ce dépôt régulièrement au cours du projet, le code final sera livré dessus.
	<p>FONCTIONNALITES DU PROJET</p>	<ul style="list-style-type: none"> • Ce festival est le vôtre, libre à vous d'inventer les noms de scènes que vous souhaitez et d'y faire passer les artistes que vous voulez (existants ou pas). • Les différentes salles de concerts et leurs états doivent être sauvegardés entre chaque exécution dans des fichiers. • Une salle est définie par son nom ainsi que ses sièges. Les sièges sont disposés en rangées. • Le nombre de rangées est choisie pour chaque salle, et chaque rangée peut avoir un nombre de sièges différent des autres rangées. • Un siège peut être réservé ou non. Il est également associé à une catégorie de prix. • En début d'exécution le programme doit proposer deux modes : <ul style="list-style-type: none"> - Le mode manager (pour les gestionnaires du festival) - Le mode festivalier (pour réserver et voir ses réservations)

- Lorsque le programme est lancé, on peut revenir à tout moment au menu de départ pour choisir entre le mode festivalier et le mode manager.
- Cela signifie que l'on peut faire n'importe quelles opérations sur les salles de concert sans jamais stopper le programme.
- Dans tous les cas, quelle que soit l'utilisation que l'on fait du programme, les allocations dynamiques doivent être bien libérées.
- De plus, à chaque modification d'une salle de concert, ou d'une réservation, tout autre événement qui modifie les données, l'ensemble des informations doit être sauvegardé sur des fichiers.

MODE MANAGER

- Le manager peut créer une ou plusieurs salles de concerts.
- Chaque salle de concert est identifiée par son nom. Le manager choisit également le nombre de rangées et de sièges par rangée. Il choisit aussi le nombre de sièges par catégorie.
- Il existe 3 catégories de sièges, la catégorie A sont les plus proche de la scène, les catégories C sont au fond et les B entre les deux.
- Les catégories sont séparées en rangées : le manager indique combien de rangées sont en catégories A, et en catégories B : rangées supplémentaires seront automatiquement en catégorie C.
- Un manager peut attribuer une salle nouvelle ou déjà existante à un concert.
- Il pourra choisir si ce concert aura une fosse au non. Si le concert a une fosse, les sièges de catégorie A sont retirés, les festivaliers seront debout. Dans ce cas l'emplacement d'un siège pourra contenir la place pour deux festivaliers. Lors de l'attribution d'un concert, le manager pourra aussi donner le prix d'un emplacement de chaque catégorie.
- Le manager définira également l'heure de fin d'un concert. Au-delà de cette heure, les sièges seront tous vidés, et la salle sera encore existante mais il ne sera plus possible de réserver de places (puisque le concert est terminé).
- Quand un concert est terminé, le manager peut modifier la configuration de la salle pour changer les prix, et les nombres de rangées des catégories A et B, peut indiquer modifier la configuration de la fosse, et modifier les prix de chaque catégorie. Enfin il pourra indiquer une date de fin de concert : ce qui permettra aux utilisateurs de réserver leurs places comme au tout début.
- Le manager peut également observer l'état des différentes salles et consulter le ratio sièges réservés / nombre total de sièges.

MODE FESTIVALIER

- Le programme affiche les salles pour lesquelles un concert est prévu et n'est pas encore terminé.
- Un festivalier peut réserver un ou plusieurs sièges (ou emplacements si jamais le concert possède une fosse) dans ces salles.
- Lors de la réservation, le programme affichera le plan de la salle en indiquant les sièges libres ainsi que leur catégorie (on pourra utiliser des codes couleurs). A l'issue de la réservation, le programme affichera le prix des places réservées.

- Exemple d'affichage de scène :

```

👉 Plan de la main stage 2: 👈
🎵 SCENE 🎵
-----
R 1 0 0 X 0 0 X X 0 0 0 0 0 0 X 0
R 2 0 0 0 0 0 0 0 0 0 X 0 0 0 0 0
R 3 0 X X 0 0 0 0 0 0 0 X X 0 0 0
R 4 X X 0 0 0 X 0 0 X 0 0 0 0 0 X
R 5 X 0 X X 0 0 0 0 0 0 X X 0 0 0
R 6 0 0 0 X 0 0 0 X X 0 0 0 0 0 0
R 7 0 0 0 0 0 0 X X 0 0 0 X 0 0 0
R 8 0 0 X 0 0 0 X 0 0 0 0 0 0 0 0
R 9 0 X 0 0 0 X 0 0 0 X 0 0 X 0 X
R10 0 0 X X 0 0 0 0 0 0 0 0 0 0 0

```

Légende:

- 0 - Catégorie A (50 euros)
- 0 - Catégorie B (30 euros)
- 0 - Catégorie C (20 euros)
- X - Siège réservé

- Des améliorations de votre choix peuvent être apportées au projet. Exemple 1 : le programme du festival est écrit à l'avance dans un fichier qui va indiquer les heures de début et de fin des concerts, et votre programme va créer les salles du programme en mémoire à partir de ce fichier. Exemple 2 : trouver automatiquement N places côte à côte dans une catégorie donnée, le plus près de la scène et le plus proche du centre des rangées.

**RESSOURCES
UTILES**

Github

- www.github.com
 - <https://docs.github.com/en/get-started/quickstart/hello-world>
- Modification des couleurs (police et fond) dans le terminal
- <http://sdz.tdct.org/sdz/des-couleurs-dans-la-console-linux.html>
- Affichage d'emoji dans le terminal
- Liste des emojis : <https://unicode.org/emoji/charts/full-emoji-list.html>