

PROJET C++ : COUPE DU MONDE

Marilyne Mafo, Maryam Aarab

MAIN 4

Année universitaire 2022/2023



Table des matières

1	Introduction	2
2	Description de My Fanzone	2
3	Organisation	2
4	Partie technique	2
4.1	Bibliothèque d'interface graphique	2
4.1.1	Notre choix	3
4.1.2	Prise en main	3
4.2	Installations requises	3
4.3	Makefile	3
4.4	Guide utilisateur	3
4.5	Architecture du code et fonctionnalités	5
5	Diagramme UML	6
6	Point de vue	8
6.1	Difficultés rencontrées	8
6.2	Fiertés	8
7	Conclusion	8

1 Introduction

Le thème "Coupe du monde" nous a évoqué beaucoup de choses qui tournent autour de la célébration lors des matchs tels que les rassemblement dans des fanzones, les feux d'artifices, les danses et les musiques créées autour de cet événement mondial. Partager ses émotions durant un match de foot procure un certain bonheur que tout le monde n'a pas souvent la chance d'avoir. Nous avons alors voulu rassembler les supporters autour de ce grand événement. C'est dans ce sens que nous avons choisi de nous orienter vers une application nommée **MyFanzone** qui permettra aux amoureux du football de trouver en région parisienne des lieux qui regroupent des supporters pour les matchs de la coupe du monde. Spécifiquement, nous avons choisi de ressasser la coupe du monde 2006 qui était épique et riche en festivités. Pour mener à bien ce projet, il sera important de répondre à plusieurs besoins et exigences qui seront explicités dans la suite.

2 Description de My Fanzone

Notre projet consiste en la création d'une application nommée "My fanzone". Celle-ci offre à l'utilisateur un choix multiple de fanzones lui permettant de visualiser le match voulu avec des personnes partageant la même passion. En utilisant l'application, il pourra avoir accès aux informations de plusieurs fanzones tel que l'adresse, la superficie, le nombre d'écran, la restaurations offertes et si l'accès est payant.

Dans le cadre de ce projet, nous avons décidé de ressasser particulièrement la coupe du monde 2006. Les fonctionnalités de l'application sont les suivantes :

- proposition de match à partir des huitièmes de finale
- proposition des fanzones où ces matchs sont diffusés
- animation, annonce du vainqueur
- esthétique de l'application (curseur en forme de ballon, texte, images...)

3 Organisation

Afin de produire un travail efficace, nous avons créé un répertoire github à l'adresse : [notregit](#) afin de partager facilement le code et les avancées. Dans le github on va retrouver un README qui explique l'objectif du projet et les procédures d'installation des packages nécessaires. En outre, toute l'architecture de notre travail qui sera explicitée par la suite.

Nous avons également utilisé *UMLET*, un logiciel qui nous a servi à produire le diagramme UML qui sera présenté dans la suite.

4 Partie technique

Dans cette partie, nous allons expliquer tout l'aspect technique/code que nous avons implémenté. Aussi, nous allons lister les différents outils nécessaires, leurs utilités et comment les utiliser.

4.1 Bibliothèque d'interface graphique

Faire une application nécessite d'avoir un aspect graphique, pour cela nous avons choisi de nous tourner vers la bibliothèque graphique *SDL2*. Cette bibliothèque contient plusieurs packages et nous en installerons quelques uns.

4.1.1 Notre choix

Dans un premier temps, en explicitant notre application, nous savions qu'il nous faudrait afficher des images, créer de boutons, jouer du sons et afficher du texte. En effectuant des recherches sur internet nous nous sommes naturellement tourner vers la bibliothèque SDL2.

4.1.2 Prise en main

Afin de bien comprendre la bibliothèques et s'assurer que nous n'avons aucun soucis à l'installation des différents packages, nous avons tout d'abord essayer d'utiliser les fonctions de base.// Nous avons commencé par créer une fenêtre au nom de "My fazone" en affichant du texte. Ensuite nous avons afficher une image, et enfin nous avons créer un bouton.

4.2 Installations requises

Avant toute compilation qui soit, il faut obligatoirement installer les packages pour la bibliothèque graphique SDL2 :

Sur Mac OS, exécuter les commandes :

1. brew install sdl2
2. brew install sdl2_image : *pour les images*
3. brew install sdl2_ttf : *pour les textes*
4. brew install sdl2_mixer : *pour le son*

Sur Linux , exécuter la commande suivante pour avoir tous les packages de SDL2 :

1. sudo apt install libsdl2-2.0-0 libsdl2-gfx-1.0-0 libsdl2-image-2.0-0 libsdl2-mixer-2.0-0 libsdl2-net-2.0-0 libsdl2-ttf-2.0-0

Ou encore, pas à pas :

1. sudo apt-get install libsdl2-dev
2. sudo apt-get install libsdl-ttf2.0-dev : *pour les textes*
3. sudo apt-get install libsdl2-image-2.0-0-dev : *pour les images*
4. sudo apt-get install libsdl2-mixer-dev : *pour le son*

4.3 Makefile

Travaillant sur MacOS et linux , nous avons remarqué qu'il fallait modifier le makefile selon le système d'exploitation mais aussi selon le chemin d'accès à la bibliothèque. Dans le dossier src, nous avons alors 2 makefile :

- Celui de MacOS qui sera dans le dossier src. Il est le makefile par défaut. Ici il faut donner le chemin d'accès à la bibliothèque SDL2 dans les variables INCLUDE et LSDL2

- Celui de Linux où les chemins d'accès ne sont pas nécessaires (c'est la seule différence avec le makefile de MacOS). Il est dans le dossier recours en cas de besoin pour ceux qui ont linux.

4.4 Guide utilisateur

Nous allons vous expliquer pas à pas comment utiliser l'application **My Fanzone**.

Etape 1 : cloner le répertoire git avec la commande git clone "lien"

Etape 2 : faire les installations requises sus-citées, vous pouvez retrouver les commandes dans le README

Etape 3 : se placer ensuite dans le dossier src et lancer la commande **make**

Etape 4 : lancer l'exécutable **./play**

Etape 5 : s'amuser avec les instructions de l'application.

Au départ, une fenêtre avec écrit "**My Fanzone**" s'affiche avec de la musique en fond, il faut cliquer sur la barre d'espace pour passer à la page suivante.

Ensuite, tous les matchs de huitièmes de finale sont affichés sur deux pages et il faut sélectionner celui que vous souhaitez regarder en cliquant dessus.

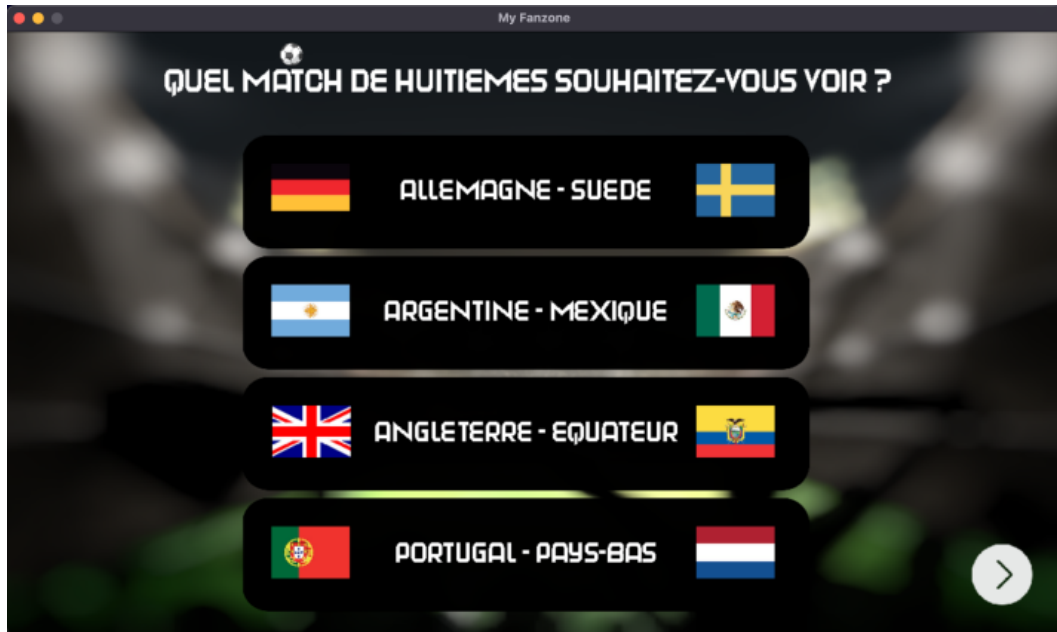


FIGURE 1 – Affichage des matchs de huitièmes de finale

Une fois le choix fait, il vous est proposé des lieux où il y aura des fanzones pour ce match sélectionné.



FIGURE 2 – Affichage des lieux de fanzone

Une fois le lieu sélectionné en cliquant dessus vous pourrez voir le gagnant étant donné que nous revivons la coupe du monde de 2006.



FIGURE 3 – Affichage des lieux de fanzone

Le procédé se répète immédiatement pour les quarts de finale, demi finale et finale.

4.5 Architecture du code et fonctionnalités

Le répertoire principal est divisé en plusieurs autres répertoires. Nous avons :

- le dossier **data** qui va contenir toutes les données dont nous auront besoin (fichiers.txt, images, sons)
- le dossier **tests** qui contient les tests unitaires.
- le dossier **umlet** qui contient les fichiers du diagramme UML
- le dossier **src** qui contient tout le code source (les classes et leurs entêtes)
- le README

Nous allons maintenant nous intéresser au dossier source src.

Pour concevoir cette application nous avons créé plusieurs classes avec des spécificités.

Approche

Tout d'abord, il a fallu créer les matchs et les lieux de rassemblement. Pour les matchs, une classe a suffit à le faire. Pour les lieux en revanche il en existe plusieurs types. Un lieu peut être intérieur ou extérieur. Parmi les lieux intérieurs on compte les restaurants, les bars (où l'on peut avoir à manger ou non) et les salles réservées. Pour les lieux extérieurs nous avons les places et les parcs. Après avoir établi nos classes et nos hiérarchies, le code s'est mis en marche.

Partie terminale

Pour gérer les matchs et leur création, nous avons la classe **Match** et la classe **LesMatch** qui contient un **vecteur** des matchs chargés en base depuis le répertoire data.

Pour gérer les lieux des fanzones, on a les classes **Lieu**, **LesLieux**, et, **Intérieur** et **Extérieur** qui héritent de **Lieu**. Les classes **Bar**, **Salle**, **Restaurant** héritent de **Intérieur** et la classe **Parc** hérite de **Extérieur**.

LesLieux contient un vecteur de lieux chargés depuis la base data.

L'utilisation des containers était utile pour le stockage de nos data. On en trouve aussi dans le fichier `main`.

Le second container est une **map** pour le stockage des drapeaux ainsi qu'une clé qui correspond à chaque drapeau. L'utilisation de la map permet d'accéder plus facilement à un drapeau surtout qu'ils sont menés à être utilisés à multiple reprise.

En outre on a une **surcharge d'opérateur** dans la classe **Lieu**. En effet, on a choisi de faire une surcharge d'opérateur pour l'affiche d'un lieu. Ceci facilite la tâche lorsqu'on les manipule et qu'on fait des tests au niveau du terminal. Au lieu d'appeler une fonction d'affichage on peut directement utiliser l'opérateur de sortie standard et c'est pratique.

Partie graphique

Pour la partie graphique, il a fallu subdiviser le code et créer des classes pour gérer différents aspects.

- la classe `Mouse` pour gérer la souris customisée en ballon de foot
- la classe `Text` pour gérer la création des surfaces pour l'écriture des textes
- la classe `Button` pour la création des boutons sur l'interface
- la classe `Image` pour charger et afficher les images
- la classe `Graphic` pour afficher les éléments graphiques

Il faut noter que les classes **Button**, **Image** et **Text** héritent de la classe **Graphic**. La classe `Graphic` contient une **méthode virtuelle** car elle est une classe abstraite. La méthode `load` permet d'afficher un élément graphique. Sachant qu'on a différents types d'éléments graphiques, la méthode est virtuelle et les classes filles vont alors redéfinir cette méthode elles-mêmes.

En plus, on a des destructeurs en méthode virtuelle.

5 Diagramme UML

Avec l'aide du logiciel UMLet, nous avons pu produire les diagrammes suivants qui établissent les différentes relations entre nos classes. Nous avons répartis en deux fichiers les diagrammes.

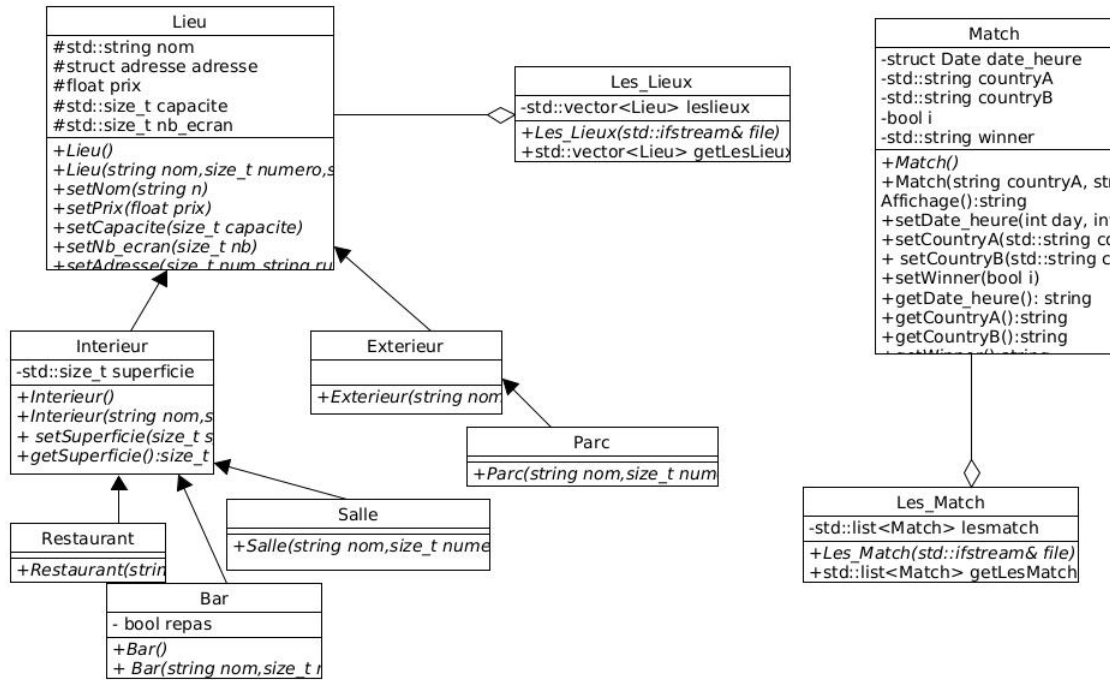


FIGURE 4 – Classes Lieu, Les_Lieux, Interieur, Restaurant, Bar, Salle, Exterieur, Parc, Match, Les_Match

Ici on a donc 3 niveaux de hiérarchie sur les lieux et les classes LesLieux et LesMatch qui incluent respectivement Lieu et Match. En effet, un lieu peut être en intérieur ou extérieur. Ensuite, on distingue différents types de lieu en intérieur ou en extérieur.

Pour la partie graphique nous avons les relations ci-dessous.

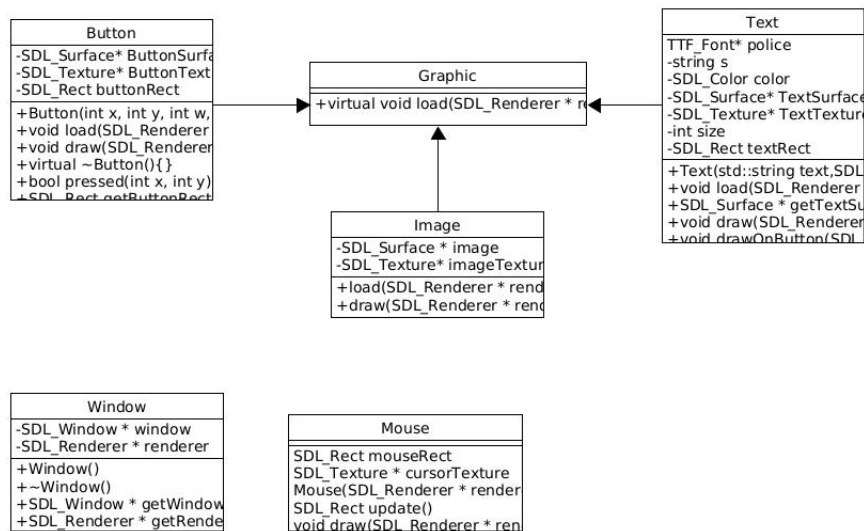


FIGURE 5 – Classes Grahic, Button, Text, Image, Window, Mouse

On voit bien que les classes Button, Image et Text héritent de la classe Graphic.

6 Point de vue

6.1 Difficultés rencontrées

Lors de ce projet, nous avons rencontré des difficultés au niveau de la prise en main et l'utilisation de la bibliothèque graphique SDL2. Il a fallu regarder beaucoup de tutoriel pour s'y retrouver. En effet, elle n'est pas très simple à utiliser lorsqu'il faut faire des longs codes. Il faut tout faire soit même.

6.2 Fiertés

La partie du projet et du code dont nous sommes le plus fières est la partie graphique. En effet malgré la complexité et la longueur des codes pour faire une action minime, une fois la bibliothèque prise en main, nous avons voulu faire un beau rendu visuel. Nous nous sommes beaucoup attardé sur cette partie et ça a pris du temps. Nous avons découpé en plusieurs sous classes la partie graphique et nous avons customisé le rendu à travers les images, les affichages, les formes et les sons. Le but étant que ce soit agréable à voir et assez distrayant.

7 Conclusion

Au terme de ce travail, nous en ressortons enrichies. En effet, ce projet nous a permis de mettre une fois de plus en pratique les notions vues au cours du semestre et les approfondir. En outre nous avons pris en main une bibliothèque inconnue et avons laissé notre créativité parler. Il a fallu faire face à des difficultés et trouver des solutions, ceci renforce l'autonomie et l'esprit de recherche. La gestion du temps a été une grande contrainte à prendre en compte aussi. Enfin, nous avons mis sur place une application sympathique et nous pensons que l'idée de My Fanzone pourrait être encore plus poussée que ce que nous avons implémenté, et cela pourrait être une application très utile pour les passionnés du football.