

Compte Rendu

Projet Coupe du Monde - C++

Thomas Korpai et Claire Wu
EI-SE
2022/2023

Sommaire

Introduction	3
Description détaillée de l'application	3
Mise en valeur de l'utilisation des contraintes	4
Diagramme UML	5
Exécution du code	5
Parties de l'implémentation dont nous sommes fiers	6

Introduction

Ce rapport présente notre interprétation du sujet “Coupe du Monde” dans le cadre de l’UE C++. Nous ferons tout d’abord une description détaillée de l’application, comprenant évidemment son fonctionnement. Nous détaillerons également l’utilisation des contraintes qui sont imposées par le sujet. Le rapport contiendra également la procédure d’installation de l’application, ainsi que le diagramme UML de celle-ci afin de montrer les rouages internes. Enfin, nous finirons en détaillant les parties de code qui nous rendent le plus fier dans ce projet.

Description détaillée de l’application

Ce code est un simulateur de coupe du monde de football. Vous contrôlez une équipe devant faire son chemin jusqu’à la finale et évidemment, jusqu’au trophée ! Cependant, ce chemin sera semé d’embûches car des événements viendront altérer votre partie aléatoirement.

Commençons par le détail de la classe Equipe. Avant tout, une équipe est composée de plusieurs joueurs et d’un coach. Alors pour pouvoir remplir nos équipes, nous avons donc d’autres classes telles que Joueur et Coach . Afin de pouvoir évaluer le niveau d’une équipe et de pouvoir la comparer avec les équipes adverses, nous avons ajouté à notre classe Equipe un attribut valeur. Cette classe permettra donc de créer votre équipe et les équipes adverses que vous allez affronter.

Afin de mieux comprendre le jeu, nous allons expliquer brièvement les classes Joueur et Coach qui sont semblables étant donné qu’il hérite tout deux de la classe Participant. Mais bien sûr, il existe quelques différences selon le rôle. Comme nous le savons, le niveau d’une équipe est représenté par les joueurs qui le constituent avec leurs points forts et leurs faiblesses. Mais dans notre jeu, les joueurs auront un niveau de potentiel pour représenter leurs points forts et pour leurs faiblesses nous aurons donc un niveau de “connerie”. Logiquement les potentiels des joueurs permettent d’ajouter de la valeur à l’équipe alors que le niveau de “connerie” diminue le niveau de l’équipe.

Quant au personnage Coach, celui-ci est représenté par son niveau expérience qui permet de rajouter un plus-value au niveau de l’équipe mais comme tout coach, il est aussi représenté par son niveau de rage qui du coup enlève de la valeur au niveau de l’équipe.

De plus, si vous avez de la chance un joueurGold pourra venir renforcer votre équipe, et si vous avez une chance monstre vous pourrez avoir même plusieurs joueurs de type Gold.

Comme dit précédemment le but de jeu est de réussir à battre les équipes adverses à travers différents matchs. Nous avons alors implémenté une classe match qui permet de gérer le déroulement d’un match. Lors d’un match, votre équipe affronte une autre équipe mais, selon votre chance, vous allez rencontrer soit des obstacles qui vous seront néfastes soit des bonus qui aideront votre équipe à gagner. Si votre bonne étoile est de votre côté, vous pourrez rapporter des bonus qui pourraient affecter les joueurs adverses ou même

marquer un but. Mais si votre étoile vous a abandonné alors ce sera votre équipe qui sera affectée ou l'équipe adverse marquera un but.

Les événements à venir vont apparaître soudainement au cours du match et vont affecter l'une des deux équipes. Mais il se pourrait que vous puissiez choisir votre destin ou le destin de l'équipe adverse : le jeu vous proposera alors de choisir entre deux choix. Les différents événements qui pourraient surgir peuvent être :

- Un but marqué
- Blessure d'un joueur
- L'intervention d'un être supérieur
- Un joueur malade

Ces différents événements vont changer le cours du jeu et pourront soit vous aider à gagner soit vous mettre en mauvaise posture.

Afin de pouvoir gérer les différents événements au cours du match, nous avons implémenté une classe "événement" mais nous avons aussi implémenté une classe "effet" qui pourra gérer les conséquences de ces événements. Cette classe "effet" va alors modifier le niveau de l'équipe selon les événements que vous allez rencontrer. Mais bien sûr, il se pourrait qu'une équipe gagne un but à travers l'un des événements.

Mise en valeur de l'utilisation des contraintes

La création du diagramme UML, nous a permis de voir l'ensemble des classes que nous devons créer pour le bon fonctionnement de notre application. De plus, cela nous a permis d'avoir une idée de ce nous voulons : les classes nécessaires mais aussi les méthodes que nous allons devoir implémenter pour mettre en place notre idée.

Les commentaires sur le code permettent à notre binôme de se retrouver dans le code de l'autre et de pouvoir continuer à sa suite. Mais aussi de nous retrouver entre deux journées ou même lorsque nous voulons revoir le code plus tard après finition du code.

Nos méthodes et fonctions ont respecté la contraintes des 30 lignes cela a permis de rendre les codes plus lisible mais aussi de fragmenter le code en plusieurs fonctions/méthodes simples, qu'on pourrait utiliser pour d'autres futur code.

Bien sûr nous avons utilisé valgrind lorsque nous avons fait une erreur de compilation que nous ne comprenons pas, nous avons pu utiliser valgrind pour plus de détails. Mais aussi cela permet de vérifier si nous avons bien désalloué tout ce que nous avons alloué dans le code.

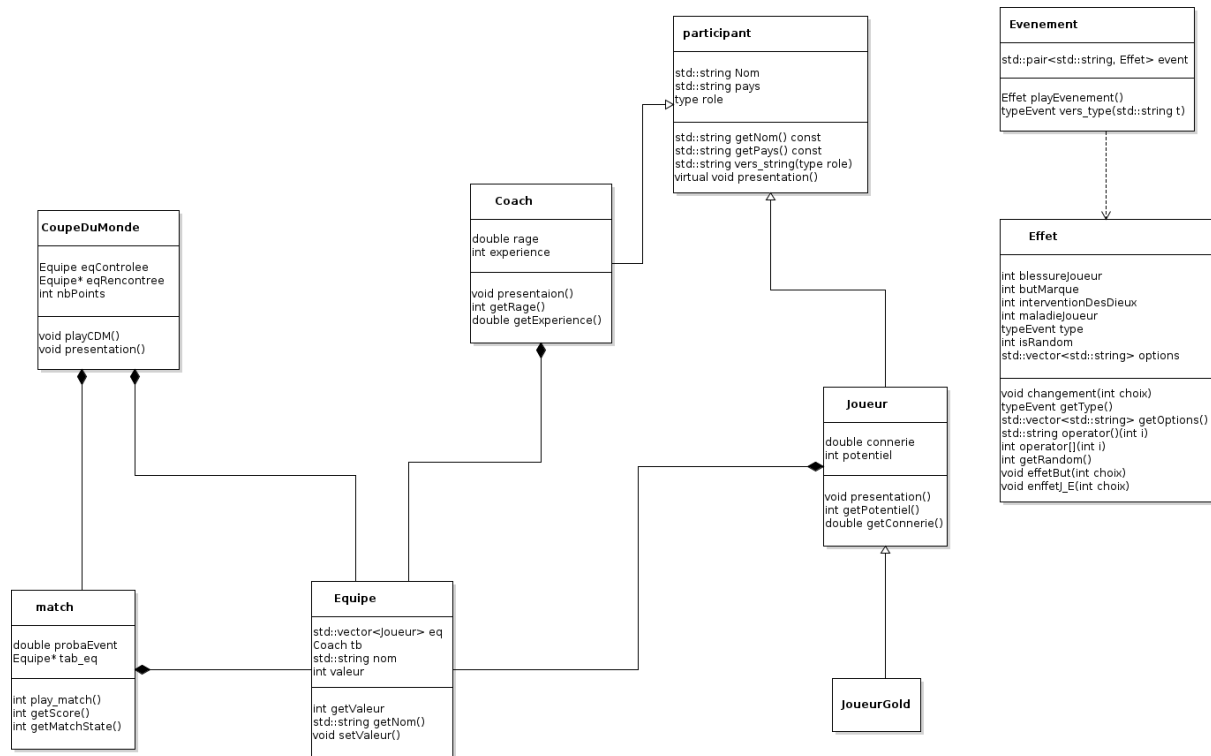
Et enfin pour faciliter le partage de nos codes, nous avons créé un git où nous mettons toutes nos progressions et corrections apportées. Cela nous permet de partager tous nos fichier mais aussi tous les changements fait dans les codes lorsque l'un des deux fait des modifications. Nous avons aussi inclus un Makefile afin de compiler plus facilement nos fichiers étant donné que nous avons plusieurs fichiers éparpillés sur plusieurs répertoires.

Différentes contraintes nous ont été imposées :

- un nombre de classes (8) que nous avons respectées dans l'organisation même du jeu
- un nombre de niveau de hiérarchie (3) qui est validé par l'organisation des différents membres d'équipes (voir UML)

- des fonctions virtuelles à utiliser : Nous avons implémenté la méthode présentation() d'un participant en virtuel. En effet, chaque type de participant possède différentes statistiques propres, ce qui signifie donc une redéfinition à chaque fois de leur présentation
- deux surcharges d'opérateurs que nous avons utilisées dans notre gestion des effets et des événements

Diagramme UML



Exécution du code

Nous avons pas implémenter d'interface graphique, vous avez juste à compiler, lancer l'exécutable et suivre les instructions du jeu sur le terminal.

Le fonctionnement est rendu simple, grâce au Makefile.

Le jeu commence avec le choix de quel équipe voulez vous contrôler :

```

1) FRANCE      2) ESPAGNE      3) ANGLETERRE
4) COREE       5) ITALIE       6) GRECE
7) ALLEMAGNE   8) MEXIQUE      9) JAPON
10) ARGENTINE  11) BELGIQUE     12) CROATIE
13) MAROC      14) PORTUGAL   15) LIBAN
16) PAYS-BAS  17) EGYPTTE      18) ALGERIE
Choisissez l'équipe que vous souhaitez contrôler :
  
```

Vous aurez ensuite la présentation des équipes qui participent à la coupe du monde :

```
Les équipes de cette coupe :  
Equipe de ESPAGNE  
Equipe de FRANCE  
Equipe de ANGLETERRE  
Equipe de ALGERIE  
Equipe de MEXIQUE
```

Et enfin nous aurons les matchs qui vont se suivre et si vous avez de la chance, vous participerez au match final pour tenter de remporter la coup :

```
Bonjour et bienvenue dans ce match qui opposera Equipe de ESPAGNE et Equipe de F  
RANCE !  
N'attendons pas plus !  
La première mi-temps commence !  
  
(Pour continuer, entrez une touche)  
m  
  
Diego Maradona revient sur Terre pour mettre un tacle à un joueur adverse  
Les dieux du foot et de la pizza vous bénissent et démontent l'autre équipe  
Temps du jeu 0 : Equipe de ESPAGNE 0: 0 Equipe de FRANCE  
  
(Pour continuer, entrez une touche)  
m  
  
Suite à l'invocation du public un dragon apparaît et fait fuir l'équipe adverse  
Les dieux du foot et de la pizza vous bénissent et démontent l'autre équipe  
Temps du jeu 1 : Equipe de ESPAGNE 1: 0 Equipe de FRANCE  
  
(Pour continuer, entrez une touche)  
m
```

Parties de l'implémentation dont nous sommes fiers

Lors de l'implémentation de notre code, nous pouvons dire que nous sommes fiers de la partie création d'événements. Avec cette partie nous avons pu laisser libre cours à notre imagination et créer des matchs avec des événement complètement loufoques. Nous n'avons pas voulu créer quelque chose de sérieux, faire quelque chose de drôle et créer des événements qu'on ne pourrait s'y attendre.

Pour finir, nous voudrions aborder ce qui nous plait le plus dans notre projet, au moyen de petits textes personnels :

(Thomas) : Pour ma part, la partie concernant les événements est ce qui me plait le plus. Si on regarde en détail, on pourrait très certainement faire mieux (avec du temps en plus) mais je trouve que ce que nous proposons dans ce projet est déjà une version bien faite, proposant des retournements de situation plutôt sympa. Le fait de simuler un match de foot et de voir arriver un événement sur un joueur qui n'a rien à voir avec le foot me fait beaucoup rire.

(Claire) : La partie que j'ai appréciée dans ce projet est la partie où on pouvait créer des événements improbables. Bien sûr, je suis fière de toutes les parties que nous avons implémentées, même si ce n'est pas parfait nous avons pu aller au bout de notre idée tout en s'amusant.