

Rapport de conception du projet programmation orientée objet

Licence d'informatique – 2ème année

Faculté des sciences et technique de Nantes

Titre du projet : Simulation de tournoi sportif.

Présenté par

Erwan PALIERNE, Daoud ABDELKERIM, Jaurès AZANDOSSESSI, Mohamed LIAMIDI
le 17-11-2023.

encadré par Laurent GRANVILLIERS.

1. Cahier des charges

. Contexte du projet

Notre projet porte sur la simulation de tournoi sportif dans un contexte général et pas de spécialisation suivant un domaine sportifs, nous ferons donc l'implémentation des différents formats de tournoi sans spécialisation spécifique.

. Objectifs du projet

L'objectif de notre projet, est de développer une application de simulation de tournoi sportif utilisant la programmation orientée objet (POO) pour offrir une modélisation réaliste et extensible.

. Fonctions applicatives

Nous aurons :

Gestion des équipes : Permettant la création des équipes, la modification des performances propre à chaque équipe, d'obtenir le nom d'une équipe ou sa performance.

Gestion des matchs : Permettant la création des matchs par deux équipes, de lancer un match, d'effectuer un tirage pour donner le vainqueur ou le perdant de chaque match.

Formats de tournoi : Qui prend en charge les différents types de formats de tournoi comme par exemple : l'élimination directe, la double élimination ou encore le format toutes rondes.

Affichage des résultats : Affiche les différents résultats des différents matchs effectués en donnant les perdants et les vainqueurs.

Interface utilisateur : Développer une interface intelligente permettant de faciliter l'interaction entre l'utilisateur et le programme.

. Cas d'utilisation

Ajouter une équipes : L'utilisateur peut ajouter une équipe aux tournoi juste en fournissant son nom et sa performance.

La gestion des équipes : L'utilisateur peut modifier le nom et même la performance d'une équipe donnée.

Configurer le format du tournoi : L'utilisateur peut choisir le format de tournoi souhaiter comme par exemple groupe de phase et élimination directe ou même juste élimination directe.

Simuler le match : Le système simule un match entre deux équipes et renvoie les résultats du match en fonction des performances et du tirage aléatoire de chaque équipe.

2. Architecture

(a) Description générale

Dans le contexte de la simulation de tournoi, nous avons utiliser plusieurs classes que sont :

Equipe : Une classe permettant de représenter une équipe participant au tournoi . Chaque équipe aura un nom et une performances données que l'utilisateur pourra entrer lui même à la création des équipes.

Match : Cette classe permet de représenter un match entre deux équipes .Elle contient donc les informations sur les deux équipes participantes, le vainqueur et le perdants à l'issu de chaque match jouer.

Tournoi : Cette classe est la classe principale qui représente le tournoi dans son ensemble, elle contiendra les différentes phases choisies par l'utilisateur pour le déroulement de son tournoi, le vainqueur final du tournoi, et la présentation des historiques du tournoi.

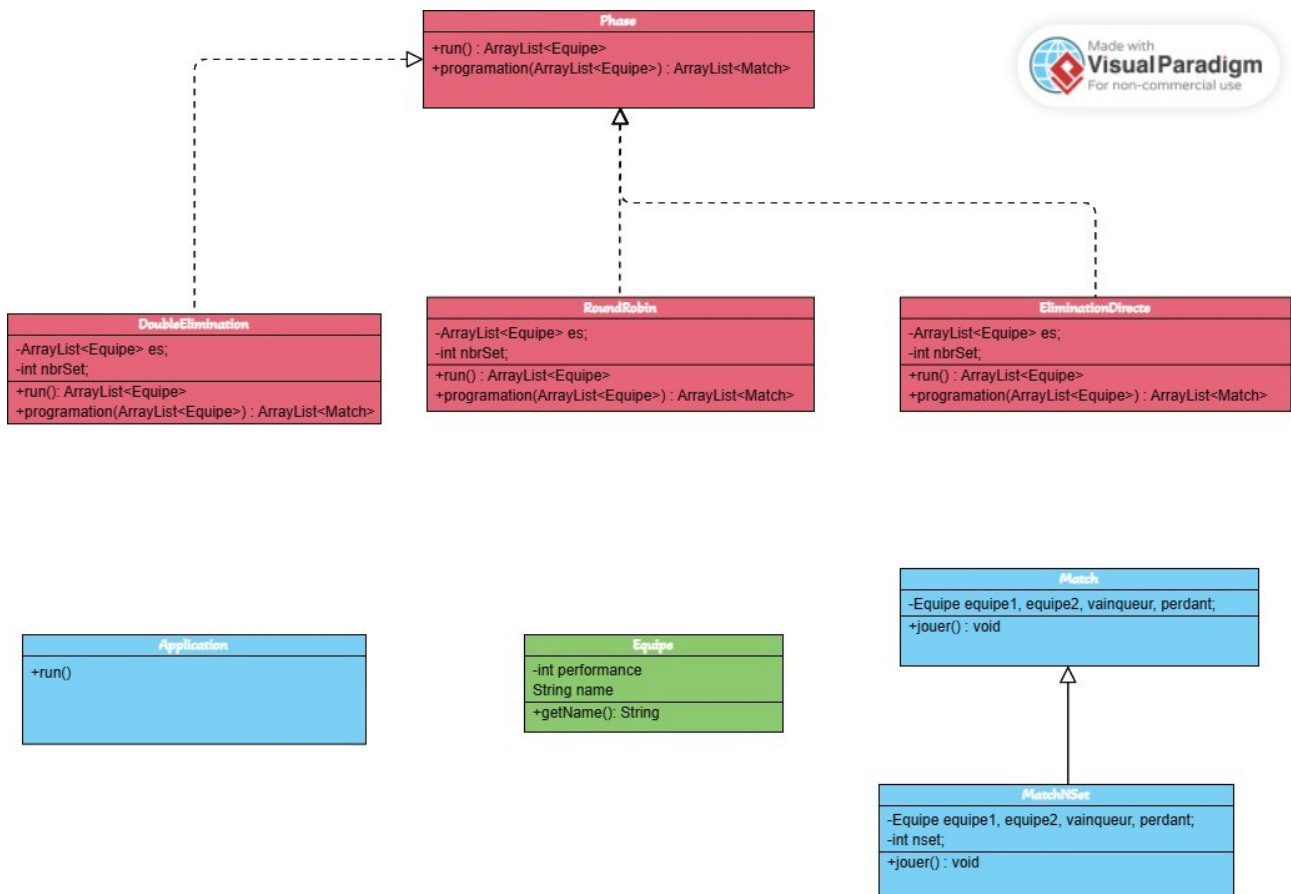
Phase : Qui est une classe abstraite dans lequel nous avons fait l'implémentation d'une méthode run et programmation dont tout les autres phases d'un tournoi héritent pour le déroulement du tournoi.

EquipeSpecial : Qui représente une équipe ayant un 0 probabilité de gagner un match. Nous avons crée une classe pour chaque phase d'un tournoi comme par exemple, une classe pour la phase de l'élimination directe, la phase pour la double élimination.

Relations entre les classes

Nous avons une relation d'héritage entre la classe **Phase** et les classes d'implémentation des différentes phases d'un tournoi, par exemple la classe EliminationDirecte hérite des méthodes run et programmation de la classe abstraite **Phase**. Nous avons aussi, une relation d'héritage entre la classe **Match** et la classe **MatchNSet**. Le fait d'avoir plusieurs implémentation des méthodes run et programmation dans les différentes classes, nous avons ainsi une relation de polymorphisme entre les classes. Au niveau de l'interface graphique, nous avons l'utilisation de plusieurs objets différents a travers une interface commune qui fait appel au polymorphisme d'héritage.

(b) Diagramme de classes



(c) Interfaces

Par rapport à l'interface, nous avons décidé de réaliser une interface graphique si l'on arrive à le faire et dans le cas contraire se limiter juste à une interface console. Nous avons pensé, à une interface comportant une fenêtre principale pour commencer expliquant un peu comment va se dérouler notre simulation de tournoi, une description bref du contenu, le titre serait donc la simulation de tournoi. Un bouton vous permettant d'être redirigé directement vers la fenêtre de rajout des équipes, vous demandant de saisir les équipes et les performances avec ajouts à la suite, vous présentant ensuite les équipes après l'ajout total avec les performances de chaque équipe. Spécifiquement, nous avons convenu utiliser des performances d'équipe compris entre 0 et 1 ce qui signifie que nous envoyons un message d'erreur lorsque l'utilisateur nous renvoie une performance ne respectant pas notre condition de jeu, ce qui permet de le guider tout au long de la simulation. Nous aurons un panneau des phases pour faire le choix de la manière dont la manière dont souhaitera l'utilisateur de faire déroulé son tournoi. Nous aurons une partie pour faire le choix en fonction du nombre de set désiré pour le tournoi en lui même . Nous aurons après tout cela, la fenêtre du tournoi même qui affichera chaque match jouer, les résultats de chaque match ou le vainqueur à la fin de chaque set dans le choix d'un set supérieur à 1. Nous vous présenterons donc les résultats finaux du tournoi selon les combinaisons des phases choisies. Tout ceci représente donc une description de notre interface graphique qui peut peut-être avoir quelques retouches lors de sa conception.

(d) Aspects spécifiques

Description d'aspects spécifiques au sujet du projet s'il y en a, par exemple des algorithmes non triviaux

Dans le cadre de notre simulation de tournoi nous nous sommes posés plusieurs questions par rapport à la gestion de la saisie par l'utilisateur d'un nombre impair d'équipe. Pour ce faire donc, nous avons décidé de créer une classe spécial nommé **EquipeSpecial** qui hérite d'équipe mais avec une performance nulle et sans nom. Vu que notre tirage de vainqueur se fait de manière aléatoire en fonction de chaque performance des équipes, notre équipe spéciale n'a donc aucune chance pour gagner un match. Ce que nous permet donc de gérer le cas de la saisie de nombre impair par l'utilisateur car à chaque fois que nous aurions un ajout de nombre impair nous rajouterons une équipe spécial qui équilibre le nombre d'équipes.

Pour gérer les différentes phases des tournois, nous avons jugé bon de créer une nouvelle classe abstraite phase, dont toutes les phases d'un tournoi hérite qui facilite donc l'implémentation des différentes phases du tournoi.

3. Regard critique

****Sur la mise en œuvre des concepts de la POO***

Les concepts de la POO nous a permis de facilement implémenter toutes les différentes parties de notre projet. En effet, chaque classe représente un module indépendant d'un système de simulation de tournoi tel que les classes EliminationDirecte, DoubleElimination, etc.

****Sur le calibrage du projet et la quantité de travail fournie***

Le projet a demandé une importante quantité de temps et de travail ; nous devions établir de nous-mêmes l'organisation du programme. Heureusement, nous avons obtenu lors des TP encadrés des conseils et des réponses à nos questions, nous permettant d'éviter des erreurs et de résoudre les difficultés rencontrées.

****Sur le travail, la motivation et l'organisation de l'équipe***

L'intégralité du groupe fut motivé, le travail demandé a permis de développer les compétences de chacun. De plus, nous organisons des réunions chaque semaine pour discuter d'éventuelles points plus complexes.

****Sur les compétences de l'équipe au début du projet et le développement de nouvelles***

Les membres de notre équipe disposent de compétences distinctes, le fait de travailler en groupe nous a permis de s'inspirer de nos partenaires. À présent, nous essayons d'apprendre de nouvelles compétences pour développer des interfaces visuelles et graphiques pour mener à bien notre projet.

Sur des perspectives d'extension du projet et sur l'extensibilité de la solution actuelle

Par rapport, à la perspectives d'extension nous avons par exemple créer une classe abstraite **Phase** pour permettre l'implémentation facile lors de l'ajout de nouvelles phases de tournoi . Dans notre structure nous avons une classe **EquipeSpecial** qui est une équipe ayant une probabilité à 0 d'être vainqueur à l'issu d'un match, il nous permet donc de gérer les cas où l'utilisateur nous renvoie un nombre impair d'équipe, ce qui veut donc dire que notre structure de simulation de tournoi est adaptée à n'importe quel nombre d'équipe participant au tournoi. Par exemple, lors d'un tournoi avec des équipes de nombre impairs nous avons l'ajout d'une équipe Spécial en fait qui est une équipe qui perd forcément chaque match joué, le vainqueur à l'issu d'un match pareil est toujours l'équipe affrontant une équipe spéciale.