



PROJET DE P00 DE L2 INFORMATIQUE

Simulation de tournoi sportif

Présenté par : AZANDOSSESSI Jaurès, ABDELKERIM
Abdel, PALIERNE Erwan, LIAMIDI Deen.

Encadré par: GRANVILLIERS Laurent.

Le 19/12/2023



INTRODUCTION

Bienvenue dans notre projet de simulation de tournoi sportif en programmation orientée objet. Ce projet a pour ambition de créer une application permettant la gestion et la simulation de tournois sportifs. À travers l'utilisation de concepts clés de la programmation orientée objet, notre application offre une expérience immersive aux utilisateurs, leur permettant de configurer des tournois, créer des équipes, définir des règles et suivre le déroulement des compétitions. Suivez-nous dans cette aventure de codage où le monde du sport rencontre la puissance de la programmation. Que le tournoi commence !



CONTEXTE

Notre projet de simulation de tournoi se distingue par son approche globale, plaçant la modélisation et la simulation d'événements sportifs au cœur de son développement, à travers les principes de la programmation orientée objet. L'objectif central qui a guidé notre démarche était de concevoir une application flexible, capable de gérer divers tournois sportifs, et offrant ainsi une expérience de simulation complète.

Au sein de notre architecture, plusieurs classes fondamentales jouent un rôle essentiel, chacune contribuant de manière significative à la richesse et à la flexibilité de notre simulation. Notamment, la classe **Equipe** se charge de définir une équipe en fonction de son nom et de sa performance, évaluée sur une échelle de 1 à 10.

La classe **Match** intervient pour créer des rencontres entre deux équipes, tandis que sa sous-classe, **MatchNSet**, hérite de ses fonctionnalités pour gérer des matchs incluant des sets. La méthode cruciale ici est **jouer**, déclenchant le déroulement du match à travers un tirage aléatoire pour déterminer le vainqueur. Nous approfondirons ultérieurement notre méthode de tirage des vainqueurs pour chaque match

L'interface `Phase`, conçue comme une interface générique, est implémentée par quatre phases distinctes de notre tournoi : élimination directe, double élimination, round-robin et phase de groupe. Chacune de ces phases est concrétisée par des classes dédiées, apportant ainsi sa propre logique de fonctionnement au déroulement du tournoi.

Concernant l'analyse statistique, nous avons introduit deux classes abstraites : **Statistiques** et **StatistiquesNSet**. Ces classes servent de modèles aux classes qui en héritent, permettant le stockage méticuleux des statistiques de chaque phase du tournoi.

Enfin, pour assurer une interaction fluide avec notre application, les classes **Interface**, **FenetreTournoi** et **SimulationTournoi** ont été développées. Ces composants contribuent à une expérience utilisateur intuitive grâce à une interface graphique soigneusement élaborée.

Notre méthode de tirage repose sur un processus purement aléatoire, exploitant les performances individuelles des équipes pour déterminer le vainqueur d'un match. Voici comment ce tirage est orchestré :

1. Pour chaque match, nous générons une liste en ajoutant un nombre de zéros équivalent à sa performance. Par exemple, si l'équipe 1 a une performance de x , nous ajoutons x zéros à la liste.
2. Ensuite, nous faisons pareil avec l'équipe 2 mais en ajoutant cette fois si des 1.
3. Cette liste ainsi constituée crée une distribution de probabilités, où chaque équipe a une chance relative de succès proportionnelle à sa performance.
4. En utilisant la méthode ``random`` de Java, nous choisissons un indice au hasard dans la liste.
5. Si le nombre à cet indice est '0', l'équipe 1 est déclarée vainqueur du match. Sinon, l'équipe 2 est désignée comme la gagnante.

Cette approche garantit que chaque équipe a une probabilité juste de remporter le match en fonction de sa performance. Ainsi, le tirage aléatoire reflète les performances respectives des équipes, apportant une dose de réalisme à la simulation du tournoi.

RÉALISATION TECHNIQUE

Lors de la conception de notre application, nous avons délibérément opté pour une interface graphique afin d'offrir une expérience utilisateur plus conviviale. Nous avons utilisé la bibliothèque Java Swing pour créer une interface visuelle interactive et intuitive.

Fenêtre d'Ajout des Équipes : L'une des caractéristiques principales de notre interface graphique est la fenêtre d'ajout des équipes. Cette interface permet à l'utilisateur de saisir les détails de chaque équipe participant au tournoi, y compris le nom de l'équipe et sa performance. Cette fenêtre offre une interaction conviviale, simplifiant le processus d'inscription des équipes de manière visuelle.

Choix du Type de Tournoi : Le choix du type de tournoi est directement intégré à l'interface. L'utilisateur peut sélectionner le format du tournoi parmi différentes options, telles que l'élimination directe, la double élimination, le round-robin, ou la phase de groupe. Cette flexibilité permet à l'utilisateur de personnaliser son expérience de tournoi en fonction de ses préférences.

Sélection du Nombre de Sets : La sélection du nombre de sets est gérée de manière interactive grâce à une boîte de dialogue de la bibliothèque Java Swing. Après avoir choisi le type de tournoi, une boîte de dialogue s'affiche, permettant à l'utilisateur de définir le nombre de sets par match.

Cette approche offre une solution visuelle et pratique pour personnaliser les paramètres du tournoi en cours. En résumé, notre choix d'une interface graphique avec Java Swing vise à rendre l'expérience utilisateur aussi agréable que possible. La fenêtre d'ajout des équipes, le choix du type de tournoi intégré à l'interface, et la sélection du nombre de sets à l'aide d'une boîte de dialogue contribuent à une interaction fluide et intuitive, offrant à l'utilisateur un contrôle complet sur la configuration de son tournoi sportif simulé.

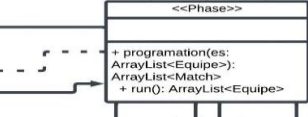
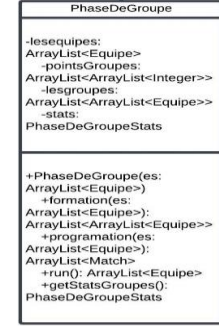
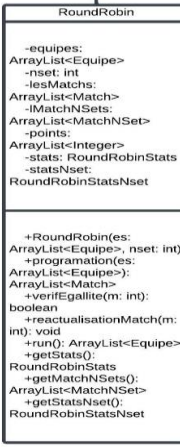
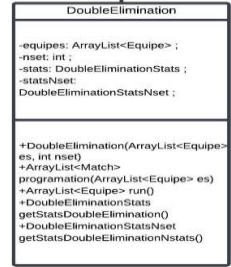
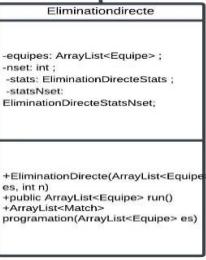
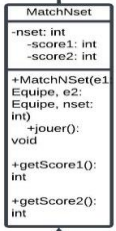
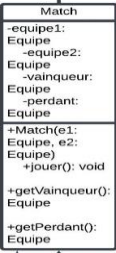
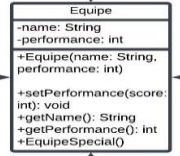
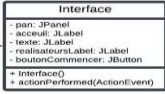
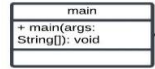
Comme vous pouvez le voir sur l'image qui suit. Vous trouverez dans le slide à suivre notre diagramme général pour notre projet.

Simulation de tournoi

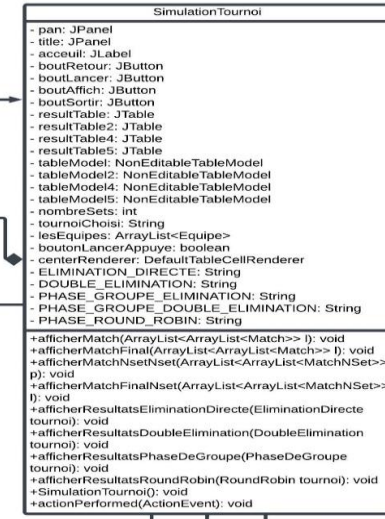
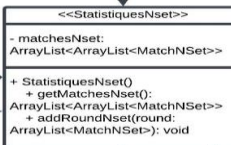
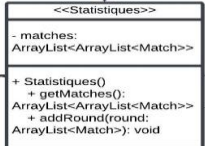
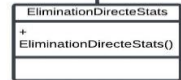
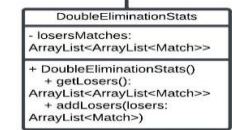
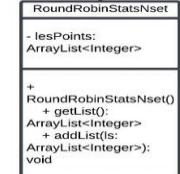
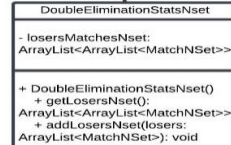
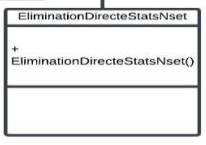
Bienvenu dans la partie d'ajout des caractéristiques de votre tournoi

Nom de votre équipes: Sa performance: Phase de tournoi:

AFFICHAGE DES EQUIPES CONSTRUITS A BASE DE VOS INFORMATIONS ENTREES



Implements





GESTION DU PROJET

Pour la réalisation de notre projet de simulation de tournoi, nous avons adopté une approche collaborative, exploitant les forces individuelles de chaque membre de l'équipe. Notre objectif était de répartir les tâches de manière à ce que chacun puisse contribuer efficacement à des parties spécifiques du projet, tout en maximisant l'expertise de chacun.

Répartition des Tâches :

Chaque membre de l'équipe s'est vu attribuer des responsabilités en fonction de ses compétences et de ses préférences. Par exemple, Jaurès AZANDOSSESSI a pris en charge le développement de la classe pour la phase de double élimination, tandis que LIAMIDI Deen s'est concentré sur la réalisation de la classe pour la phase d'élimination directe. D'autres membres de l'équipe ont également assumé des rôles spécifiques en fonction de leurs compétences respectives.

Collaboration sur l'Interface Graphique :

Concernant l'interface graphique, nous avons adopté une approche d'auto-apprentissage. Chacun de nous a entrepris des recherches individuelles sur les composants offerts par la bibliothèque Java Swing, tels que les **JLabel**, les **JTextField**, et d'autres éléments graphiques. Nous nous sommes réunis régulièrement, que ce soit pendant les travaux pratiques ou en dehors, pour partager nos découvertes, discuter des meilleures pratiques et intégrer ces connaissances dans le développement de notre interface graphique.

Collaboration Continue :

L'esprit d'équipe a été renforcé par des sessions de travail collaboratives, où les idées de chacun étaient mises en commun. Les discussions ont permis d'enrichir la conception et de garantir une cohérence globale dans le projet. Les membres de l'équipe ont été encouragés à poser des questions, à partager leurs connaissances et à résoudre ensemble les défis rencontrés.

Par la suite, nous avons décidé de travailler sur une plateforme appelée **Replit** pour faciliter la collaboration sur le projet. Cependant, après une période d'essai, nous avons constaté que cette approche entraînait des compilations erronées dues à des erreurs générées par d'autres codes. Cela a eu un impact sur l'avancement du projet et a créé des difficultés pour maintenir une progression fluide.

En réponse à cela, chaque membre de l'équipe a préféré revenir à une approche de travail sur son propre ordinateur. Cette décision a été motivée par le désir d'éviter les complications liées aux erreurs de compilation et de garantir une meilleure productivité individuelle.

Cette transition vers un environnement de développement individuel a permis à chacun de se concentrer sur son propre code, réduisant ainsi les conflits potentiels et les retards causés par des erreurs externes. Bien que la collaboration sur une plateforme unique puisse offrir des avantages en termes de visibilité et de partage en temps réel, le choix de travailler localement a été privilégié pour assurer une exécution plus fluide du projet.

BILAN TECHNIQUE

Objectifs Remplis :

- Implémentation réussie des différentes phases du tournoi, y compris l'élimination directe et la double élimination.
- Conception et réalisation de l'interface graphique permettant une interaction conviviale.
- Utilisation de la bibliothèque Java Swing pour créer une interface visuelle.

Nouvelles Technologies Apprises :

- Maîtrise approfondie de Java Swing pour le développement de l'interface graphique.
- Application des concepts de gestion de projet dans un contexte de développement logiciel.

BILAN HUMAIN

Bilan Humain :

- **Nouvelles Compétences Acquises**

Amélioration des compétences en programmation orientée objet, collaboration efficace avec une équipe, résolution de problèmes en groupe.

- **Succès Obtenu**

Réalisation d'un projet complet et fonctionnel malgré des défis techniques.

Difficultés Rencontrées :

- Complications liées à la collaboration sur une plateforme unique (**Replit**), résolues en optant pour un développement local.



AUTO-ÉVALUATION

Le projet de simulation de tournoi a été globalement réussi, démontrant une compréhension approfondie des principes de la programmation orientée objet et de la conception de logiciels. Les objectifs techniques ont été atteints, notamment la mise en œuvre réussie des phases du tournoi et le développement d'une interface graphique conviviale en utilisant Java Swing. Les membres de l'équipe ont acquis de nouvelles compétences techniques, en particulier dans la manipulation de Java Swing et la gestion de projet.

La répartition des tâches a été efficace, permettant à chaque membre de se spécialiser dans des parties spécifiques du projet. Cependant, des ajustements ont été nécessaires en cours de route, soulignant l'importance de l'adaptabilité dans la gestion de projet.

L'architecture du projet était globalement cohérente, bien que certaines classes puissent bénéficier d'une décomposition supplémentaire pour améliorer la modularité. La documentation du code était claire, bien que des commentaires plus descriptifs puissent être ajoutés pour expliquer la logique complexe.

La collaboration au sein de l'équipe a été positive, bien que des améliorations puissent être apportées à la gestion de la communication et à la rétroaction entre les membres. L'auto-évaluation a souligné la maîtrise technique, la gestion de projet efficace et la qualité du code.

En résumé, le projet a été un succès global, mais des opportunités d'amélioration ont été identifiées pour renforcer la collaboration, la documentation et l'optimisation du code.

PLACE À LA PRÉSENTATION DU PROJET EN LUI MÊME



CONCLUSION

En conclusion, notre projet de simulation de tournoi sportif a été une expérience enrichissante qui a permis à notre équipe de mettre en pratique et d'approfondir ses compétences en programmation orientée objet. Nous avons réussi à atteindre nos objectifs techniques, en développant une application fonctionnelle avec différentes phases de tournoi et une interface graphique conviviale grâce à Java Swing.

Ce projet nous a offert l'opportunité d'acquérir de nouvelles compétences techniques, notamment dans la manipulation de bibliothèques graphiques et la gestion de projets complexes. La répartition des tâches a permis à chaque membre de se spécialiser tout en favorisant la collaboration au sein de l'équipe.

Malgré les défis rencontrés, tels que des ajustements dans la planification et des complications liées à la collaboration sur une plateforme unique, nous avons démontré notre capacité à nous adapter et à surmonter ces obstacles.

L'auto-évaluation a souligné nos réussites en matière de maîtrise technique, de gestion de projet et de qualité du code. Cependant, elle a également révélé des domaines d'amélioration potentiels, notamment dans la communication au sein de l'équipe et la documentation détaillée du code.

Ce projet a été l'occasion de consolider nos connaissances et compétences, et il représente une étape importante dans notre parcours académique en programmation. Nous sommes fiers du résultat obtenu et des enseignements tirés de cette expérience collaborative, et nous sommes prêts à appliquer ces acquis dans nos projets futurs.

NOUS SOMMES À LA FIN DE NOTRE PRÉSENTATION

DES QUESTIONS?

