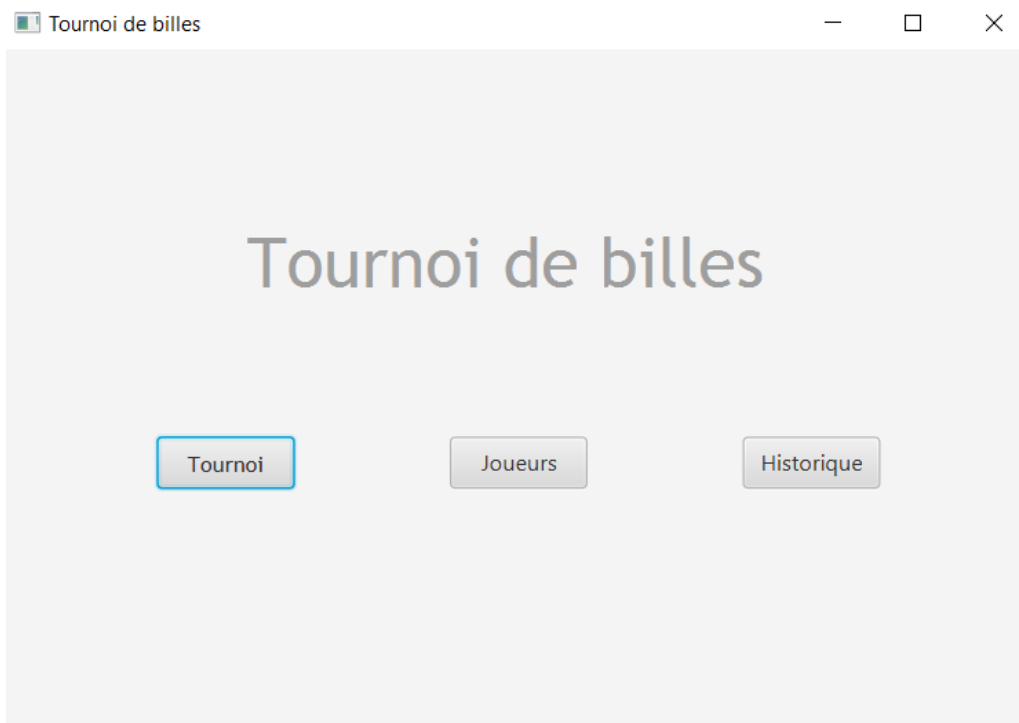


Projet Java fx : Tournoi de billes



Présentation du projet

Au cours de cette Licence 3 MIAE, nous avons appris à programmer en Java, afin de mettre en pratique nos compétences acquises, nous devons faire un projet Java fx sur l'organisation et la gestion d'un tournoi. Le choix du contexte du tournoi étant au choix, nous avons choisi un tournoi de billes.

Notre projet contient plusieurs grandes parties, tout d'abord les joueurs. Nous sommes partis du constat que 1 joueur = 1 bille. Chaque bille a plusieurs caractéristiques : *POWER*, *SPEED* & *HEIGHT*. Ces valeurs sont prises au hasard entre 0 et 10 à la création du joueur. On retrouve donc dans la partie joueur une liste de tous les joueurs existant qui sont sauvegardés dans un fichier *Joueurs.csv*. On peut ajouter un joueur en spécifiant seulement son nom. On peut afficher les caractéristiques dites *stats* de sa bille et supprimer un ou plusieurs joueurs. Afin que ce ne soit pas toujours les mêmes joueurs qui gagnent s'ils ont les meilleurs stats (10 sur power speed et height) nous avons fait en sorte que lors de l'ouverture de la section Joueurs, les stats des joueurs sont modifiés aléatoirement, toujours entre 0 et 10.

La seconde partie est la création de tournois. Ceux-ci sont avec 8 joueurs que nous choisissons parmi les joueurs créés. On donne ensuite un nom au tournoi et un type. Le type de tournoi est *ELIMINATION* & *LEAGUE* qui sont différents :

- *ELIMINATION* dit arbre de tournoi : il y a plusieurs tours, le premier affronte un autre joueur, ensuite celui ayant gagné le premier match affronte celui ayant gagné le deuxième et ainsi de suite. Afin de savoir qui gagne, à la création un type de circuit est pris au hasard correspondant aux stats des billes sur lequel les joueurs s'affronteront. Celui ayant la meilleure stat correspondant au circuit gagnera. En cas d'égalité ce sera toujours le second joueur qui gagnera. *Exemple : si c'est le circuit WEIGHT qui est choisi, le joueur qui gagnera sera celui ayant la meilleure stat de WEIGHT.* En plus de connaître le premier et second, un match sera fait pour la troisième place qui sera affichée ;
- *LEAGUE* : chaque joueur affronte tous les joueurs donc 7 matchs chacun au total. Afin de changer par rapport à l'autre type de tournoi, en ligue on reprend un circuit pris aléatoirement à chaque match. Une victoire donne 3 points, une égalité 1 point et une défaite 0 point. Le joueur ayant le plus de point gagne le tournoi.

La dernière partie est l'historique des tournois. Dans cette section on peut voir le nom du tournoi, le type de circuit associé ainsi que le podium des 3 premiers. On peut supprimer les tournois si on le souhaite.

Fonctionnalités

Navigation simple entre les différentes scènes

Création de joueurs

Affichage des stats des joueurs

Suppression des joueurs

Création de deux type de tournoi : LEAGUE & ELIMINATION

Manipulation des match réalisés

Observation des tournois réalisés

Suppression des tournois réalisés

Difficultés rencontrées

Lors de ce projet java fx, les principales difficultés rencontrées étaient sur les spécificités de java fx. Chaque nouvelle découverte de fonctionnalité amena un nouveau problème : *création et changement de scène, tableau avec les listView et tableView*. Il a alors fallu faire de nombreuses recherches, il y avait à chaque fois plusieurs solutions possibles, nous avons choisi celles faisant un code propre et compréhensible, tout en étant en harmonie avec le reste du code. D'un point de vue conception il n'y a pas eu de problème, nos alternances en entreprises nous ont permis de développer notre esprit logique.

Hors programmation pure, nous avons rencontré des difficultés avec la création de l'exécutable et les plugins pour le lancement du projet car comprendre quelle dépendance à quel rôle n'est pas chose facile quand on ne l'a jamais fait. Cependant, Maven est très efficace et nous aide énormément.

Nous avons eu une bonne gestion du groupe, chacun a eu son rôle. ONEDA Mathis à la conception, structure du projet et écriture du début de classes, DELMAS Emmanuel à la création des scènes avec le design des fichiers fxml, NONNE Matthieu à la programmation globale du projet.

Bilan outil

Ayant utilisé les outils nécessaires pour ce projet en entreprise (GIT, Maven & IntelliJ) nous n'avons pas rencontré de difficulté car nous avons déjà acquis les bases. Cependant, java fx et la création de l'exécutable demandent de modifier la structure du projet, domaine où ONEDA Mathis est compétent et qui a pu résoudre ce problème.

Conclusion

Ce projet nous a permis de mettre en pratique les cours suivis en Java cette année. De plus, le système d'interface Java fx est très intéressant et nous pourrons nous en servir pour des projets personnels. Bien que ce projet ait demandé de se former sur une technologie, il a pour nous été agréable pour deux grandes raisons : le sujet du projet est simple et intéressant et nous avons beaucoup de temps pour le réaliser nous permettant de nous focaliser dessus lors de temps libre.

Lorsque nous sommes en entreprise ce n'est pas vraiment nous qui choisissons comment organiser toute l'équipe mais dans ce petit groupe nous avons pu mettre à profit nos cours et notre expérience en entreprise pour nous organiser de manière efficace.

Nous avons lors de ce projet repris des concepts vus en entreprise avec le MVC et nous nous sommes améliorés en programmation Java, avec un code propre et compréhensible et une plus grande rapidité d'analyse des besoins.

Si le jar ne se lance pas. Lancer le projet avec la flèche verte run. Puis aller dans l'onglet Maven -> Plugins -> javafx -> Javafx:run

<https://github.com/Matthieu-Nonne/Project-Java/tree/develop/ProjectJava%20v1%20Groupe%204>