

Guide du Programmeur

Projet “Simulation de tournoi sportif”

DAMODARANE Jean-Baptiste & ELUMALAI Sriguru - Licence 3 Informatique - B

Ce document est conçu pour les développeurs qui souhaitent améliorer ou modifier le simulateur de tournoi sportif. Il explique l'algorithme utilisé ainsi que les structures de données.

I. Algorithme du programme

Il s'agit d'un tournoi où chaque compétiteur joue un certain nombre de matchs et est éliminé du prochain tour après avoir obtenu un score bas au tour actuel. Pendant chaque tour, chaque équipe joue un match contre une autre équipe, sélectionnée selon le nombre de points accumulés. Les scores des matchs sont simulés aléatoirement.

A la fin du tournoi, les scores finaux de chaque équipe sont affichés, avec une vérification pour s'assurer qu'aucun score n'était inférieur à 0. Enfin, l'équipe ayant le plus grand nombre de points est déclarée comme gagnante du tournoi.

Le programme utilise pthreads pour simuler des matchs en parallèle. Pour chaque match, un nouveau thread est créé en utilisant pthread_create, et les résultats sont obtenus via pthread_join. Le pthread_mutex_t de la structure de données Tours, permet de synchroniser l'accès aux matchs.

La fonction sleep est également utilisée pour simuler un temps de pause entre les différentes manches.

II. Structures de données

Le programme utilise deux structures de données :

- Match : représente un match entre deux équipes, avec les scores des deux équipes et leurs identifiants. Elle est utilisée pour stocker les informations d'un match dans le système.
- Tours : représente un tour de compétition, avec un tableau de Matchs qui ont lieu pendant ce tour. La structure contient également un mutex pour synchroniser l'accès concurrent aux matchs du tour. Elle est utilisée pour stocker les informations de chaque tour de compétition dans le système.

III. Fonctions principales

Les principales fonctions utilisées dans le simulateur de tournoi de football sont :

- `gener_score_random` : cette fonction permet de générer un score aléatoirement pour une équipe spécifique.
- `determiner_gagnant` : cette fonction permet de déterminer le gagnant en fonction des scores des 2 équipes qui ont participé au match.
- `simuler_match` : cette fonction permet de simuler un match avec 2 scores aléatoires pour chaque équipe et de déterminer le gagnant en appelant la fonction précédente.
- `calcul_score_verif_elimination` : cette fonction permet de calculer le score total et de vérifier l'élimination d'une équipe spécifique.
- `recherche_equipes_eliminees` : cette fonction permet de rechercher les équipes pour un tour donné et de mettre à jour les scores des équipes qui restent.
- `simuler_tour` : cette fonction permet de simuler un tour avec les match qui vont avec lui.
- `init_simuler_tours` : cette fonction permet d'initialiser les équipes et les tours, puis simule les tours et ensuite, elle met à jour les scores.

Le reste des fonctions permettent d'ouvrir et de fermer le fichier "equipes.txt" (fichier contenant les équipes), de mener des vérifications dans ce fichier, d'ouvrir et de fermer le fichier "resultats.txt" (fichier dans lequel seront écrits les résultats finaux du tournoi).

IV. Documentation en détail du contenu de programme

Pour connaître plus en détail les différentes fonctions et les structures de données utilisées, nous avons inclus la documentation générée par Doxygen. Pour cela, vous devez vous placer là où se trouve le fichier "tournoi.c", installer "sudo apt-get install doxygen" sur le terminal, si ce n'est pas déjà fait, puis taper "doxygen -g Doxyfile" (pour créer un Doxyfile) et ensuite taper "make doc". Vous verrez ensuite, l'apparition d'un dossier nommé "html" avec des fichiers html contenant des détails sur ces fonctions.

V. Conclusion

Le programme n'est pas limité aux tournois de football, mais peut être modifié et amélioré pour simuler d'autres types d'événements sportifs (à respecter les algorithmes et les structures de données). En plus, le fait d'exploiter des entrées utilisateur pour les

équipes et les scores, plutôt que de les stocker dans un fichier, il serait possible de personnaliser le tournoi selon des préférences et besoins.

Des fonctionnalités pourraient être incluses comme la création d'une interface graphique, implémentation d'un système d'analyse de données pour identifier les équipes les plus performantes et implémentation d'un algorithme basé sur la machine learning pour la simulation de score.

Répartition des tâches :

	Jean-Baptiste	Sriguru
Manipulation et configuration des fichiers textes d'entrée et de sortie	70%	30%
Simulation des matchs et génération des scores	50%	50%
Détermination de vainqueurs	30%	70%
Manipulation et gestion des équipes éliminées	70%	30%
Vérifications des scores finaux de chaque équipe	60%	40%
Simulation et manipulations des tours du tournoi	70%	30%
Fonction principale	50%	50%
Makefile	70%	30%
Documentation Doxygen	70%	30%