Université de Paris **Langages à objets avancés**
M1 **Année 2021—2022**

TP Noté n° 1

Ce travail est à réaliser chez vous et individuellement et à déposer sur Moodle au plus tard le mercredi 20 octobre 2021 à 23h59 sous forme d'une archive (utiliser impérativement la commande `tar` [1]).

Réfléchissez bien à ce qui doit être déclaré constant dans votre programme, à ce qui doit être passé par référence en argument de vos méthodes et fonctions.

Jeu de Morpion en tournoi

Vous allez écrire un ensemble de classes qui permettent d'organiser un tournoi de jeux de morpion [2] entre plusieurs participants. La grille du jeu de morpion aura toujours les dimensions 3x3.

Une partie de morpion se jouant à deux joueurs, un tournoi fait jouer les participants deux à deux de façon à ce que tous les participants jouent une fois exactement avec chaque autre participant.

Un joueur sera défini par son identifiant (suite de caractères) et son score (réel). Une partie entre deux joueurs se définit par sa grille de jeu. Un tournoi correspond à un ensemble de joueurs. Les joueurs participent à un tournoi et le tournoi organise les parties.

Diagramme UML

Faire le diagramme UML du jeu de morpion en tournoi. Attention, nous vous demandons de joindre à votre archive une image de ce diagramme ou un pdf (pas le fichier bouml, StarUML...).

Pour chaque classe définie ci-dessous vous devrez pouvoir afficher une instance en surchageant l'opérateur `<<`.

Joueur

Un joueur doit pouvoir mettre à jour son score. A l'issue d'une partie, s'il y a un gagnant, le score du gagnant doit être augmenté de la valeur $\frac{9}{\text{nbcoups+nbcoups\%2}}$ où nbcoups correspond aux nombre de coups joués lors de la partie. Le score du joueur ne fait qu'augmenter au cours du tournoi et est conservé entre deux tournois. Donc plus un joueur participe à des tournois, plus son score augmente.

1. `tar cvf mon_fichier.tar fic1 fic2 ...` pour créer l'archive `mon_fichier.tar` composée des fichiers `fic1`, `fic2`, ...
2. voir `https://fr.wikipedia.org/wiki/Tic-tac-toe` pour les règles simplissimes du jeu

Partie

Une partie doit pouvoir être lancée entre deux joueurs. La déclaration de l'attribut qui représente la grille du jeu doit se faire avec la classe `vector` de la STL.

Tournoi

Lors d'un tournoi les participants doivent pouvoir s'inscrire (`inscription` prenant en paramètre un joueur) au tournoi tant que celui-ci n'a pas débuté. Lorsque le tournoi débute, les inscriptions sont closes. Elles doivent pouvoir réouvrir à la fin du tournoi.
On doit également pouvoir désinscrire tous les participants (`vide_participant`) à la fin du tournoi.
Un tournoi doit pouvoir être organisé entre les différents participants (`lance_tournoi`) et le gagnant du tournoi annoncé.

Modalités de rendu

Nous vous fournissons un fichier `Makefile`, un fichier `main.cpp`, ainsi que des fichiers contenant des données d'entrée, `test_3J` et `test_4J`. Votre programme doit pouvoir être compilé et s'exécuter avec la commande

```
$ make test
```

Tout rendu qui ne respectera pas cette consigne aura pour note 0.

Un fichier `resultat.txt` contenant le résultat obtenu lors de l'exécution de la commande ci-dessus, vous est égalemant fourni, afin que vous puissiez vérifier que votre programme produit bien ce qui est attendu.