

## Travail pratique 1

## Échec et mat

Ce travail pratique vous permettra de mettre en application toutes les notions vues jusqu'à ce jour dans le cadre de votre cours : conception et programmation orientée objet.

### Mise en situation :

Votre premier travail pratique consiste concevoir et programmer une application permettant à deux joueurs, humains, de jouer une partie d'échec. Puisque le travail devra se faire en équipe de deux, vous prenez la décision, vous et votre coéquipier, de séparer la tâche en deux. Votre coéquipier fera l'interface utilisateur tandis que vous, le système de gestion de parties d'échec.

Après deux semaines de travail, votre coéquipier vous remet ça partie. Vous vous rendez compte qu'il n'a pas mis beaucoup d'efforts... probablement trop occupé à jouer au Hold'em. Le code de l'interface utilisateur n'est pas très optimal et pourvu de très peu de validations et de commentaires, et même d'un petit « bug » d'affichage.

Vous n'allez tout de même pas refaire son travail, vous devrez donc faire avec.



### Travail à faire :

Vous devrez développer toutes les classes nécessaires afin de gérer une partie d'échec. Ces classes devront être utilisables par l'interface de votre coéquipier afin de former une application cohérente. Vous devrez donc l'analyser quelque peu.

### Fonctionnalités de l'interface :

- À son lancement, l'interface propose une liste de joueurs triée par ordre de classement.
- Lors de la sélection d'un seul joueur, ces statistiques sont affichées
- Lors de la sélection de deux joueurs, les classements futurs sont affichés.
- Lors d'un clique sur le bouton « Jouer », la partie débute entre les joueurs sélectionnés.

- L'état de la partie en cours est toujours affiché: Coup illégal, Échec, Aux blancs à jouer, ...
- Suite à une partie, le classement des joueurs est ajusté.

#### **Système de classement :**

Si le gagnant possède un classement inférieur à son adversaire, il gagne la moitié de la différence entre les deux classements. En revanche, si le gagnant a un classement supérieur à son adversaire, il ne gagne que le quart de la différence. Du côté du perdant, s'il possède un classement supérieur à son adversaire, il perd la moitié de la différence entre les deux classements. S'il possède un classement inférieur, il ne perd que le quart de la différence.

Si la différence de classement est plus élevée que 500, le joueur possédant le plus haut classement n'augmentera pas dans le cas d'une victoire.

*Ce classement n'est **pas très équitable** et ne s'applique que dans le contexte de cette application. Le vrai système utilisé par la fédération des échecs est le système ELO. Si vous désirez l'implémenter... libre à vous.*

#### **Évaluation :**

- Bonne sélection des classes nécessaires à la réalisation du projet.
- Bonne sélection de ce qui doit être public, privé, statique et constant.
- Classes robustes retournant des valeurs valides.
- Aucun attribut, procédures ou fonctions membres et accesseurs inutiles.
- Bonne réutilisation de fonctions existantes : redéfinition et surcharge.
- Code épuré, indenté, commenté et respectant les normes de programmation.

**Date de remise :** À déterminer.