

# SPÉCIFICATION TECHNIQUE

## POUR LE DIRECTEUR DU TOURNOI SUISSE

### INTRODUCTION

Nous utilisons actuellement une application en ligne pour nous aider à gérer nos tournois d'échecs hebdomadaires. Malheureusement, cette application nous a déçus par le passé. Elle tombe souvent en panne, ce qui signifie que les matchs sont retardés. Nous organisons également des tournois dans des lieux où la connexion Internet est lente, ce qui a causé des problèmes supplémentaires.

Vous trouverez ci-dessous les spécifications techniques d'un programme autonome, hors ligne, pour console, qui répondrait à nos besoins et nous soulagerait de beaucoup de soucis.

### L'EXÉCUTION DU PROGRAMME

Nous nous attendons à ce que le programme soit écrit en Python et soit lancé depuis la console. En d'autres termes, l'exécution du programme devrait ressembler à ceci : `python <nom du programme>.py`. Le programme devrait fonctionner sous Windows, Mac ou Linux, et avoir un environnement virtuel associé.

### DÉROULEMENT DE BASE DU TOURNOI

Voici comment nous organisons généralement un tournoi d'échecs en utilisant le logiciel actuel :

1. Créer un nouveau tournoi.
2. Ajouter huit joueurs.
3. L'ordinateur génère des paires de joueurs pour le premier tour.
4. Lorsque le tour est terminé, entrez les résultats.
5. Répétez les étapes 3 et 4 pour les tours suivants jusqu'à ce que tous les tours soient joués, et que le tournoi soit terminé.

### SCHÉMA DES TOURNOIS

Chaque tournoi doit contenir au moins les informations suivantes :

- Nom
- Lieu :
- Date
  - Jusqu'à présent, tous nos tournois sont des événements d'un jour, mais nous pourrions en organiser de plusieurs jours à l'avenir, ce qui devrait donc permettre de varier les dates.
- Nombre de tours
  - Réglez la valeur par défaut sur 4.

- Tournées
  - La liste des instances rondes.
- Joueurs
  - Liste des indices correspondant aux instances du joueur stockées en mémoire.
- Contrôle du temps
  - C'est toujours un bullet, un blitz ou un coup rapide.
- Description
  - Les remarques générales du directeur du tournoi vont ici.

## PROGRAMME DES JOUEURS

Nous aimerions que l'application hors ligne contienne une base de données des joueurs. Le programme devrait avoir une section dédiée à l'ajout de joueurs, et chaque joueur devrait contenir au moins les données suivantes :

- Nom de famille
- Prénom
- Date de naissance
- Sexe
- Classement
  - Il s'agit d'un chiffre positif.

Ne vous préoccupez pas de l'importation de nos anciennes données dans le nouveau programme. Nos managers sont heureux d'entrer les joueurs manuellement lorsqu'ils arrivent aux tournois. De plus, nous n'avons pas besoin de pouvoir supprimer des joueurs.

## TOURS/MATCHS

Chaque tour est une liste de **matches**. Chaque match consiste en une **paire** de joueurs avec un **champ de résultats** pour chaque joueur. Lorsqu'un tour est terminé, le gestionnaire du tournoi saisit les résultats de chaque match avant de générer les paires suivantes. Le gagnant reçoit 1 point, le perdant 0 point. Si un match se termine par un match nul, chaque joueur reçoit 1/2 point.

Un match unique doit être stocké sous la forme d'un tuple contenant deux listes, chacune contenant deux éléments : une référence à une instance de **joueur** et un **score**. Les **matches** multiples doivent être stockés sous forme de liste sur l'instance du tour.

En plus de la liste des correspondances, chaque instance du tour doit contenir un champ de nom. Actuellement, nous appelons nos tours "Round 1", "Round 2", etc. Elle doit également contenir un champ Date et heure de début et un champ Date et heure de fin, qui doivent tous deux être automatiquement remplis lorsque l'utilisateur crée un tour et le marque comme terminé. Les instances de round doivent être stockées dans une liste sur l'instance de tournoi à laquelle elles appartiennent.

## GÉNÉRATION DES PAIRES

Nous générons des paires à l'aide du système suisse. Il devrait s'agir d'un processus automatique effectué par l'ordinateur. Veuillez consulter l'annexe pour obtenir des instructions sur l'algorithme que nous aimerions que vous utilisiez à cet effet.

## MISE À JOUR DES CLASSEMENTS

À la fin d'un tournoi, le responsable du tournoi met à jour manuellement le classement des joueurs. Tant que le programme affiche les résultats du tournoi proprement, nous serons heureux !

De plus, le gestionnaire devrait pouvoir modifier le classement d'un joueur à tout moment, et pas seulement après un tournoi.

## RAPPORTS

Nous aimerions pouvoir afficher les rapports suivants dans le programme :

- Liste de tous les acteurs :
  - par ordre alphabétique ;
  - par classement.
- Liste de tous les joueurs d'un tournoi :
  - par ordre alphabétique ;
  - par classement.
- Liste de tous les tournois.
- Liste de tous les tours d'un tournoi.
- Liste de tous les matchs d'un tournoi.

Nous aimerions les exporter ultérieurement, mais ce n'est pas nécessaire pour l'instant.

## SAUVEGARDE/CHARGEMENT DES DONNÉES

Nous devons pouvoir sauvegarder et charger l'état du programme à tout moment entre deux actions de l'utilisateur. À un moment donné, nous utiliserons probablement une base de données, mais pour l'instant, nous avons besoin de quelque chose de simple. Nous réfléchissons encore à la meilleure façon d'y parvenir de notre côté – Charlie, notre assistant informatique, vous tiendra au courant !

## VUES

Une fois le programme lancé, l'utilisateur utilisera le menu principal pour effectuer des actions. Nous vous laissons le soin de décider de la liste des menus. Tant que l'utilisateur peut effectuer les actions spécifiées ci-dessus, nous serons contents !

## LA STRUCTURE ET LA MAINTENANCE DU CODE

Le code doit être aussi propre et maintenable que possible pour éviter les bugs. Nous attendons que la structure du programme suive le modèle de conception **modèle-vue-contrôleur**. En d'autres termes, le programme devrait être divisé en trois paquets : modèles, vues et contrôleurs, mais vous pouvez ajouter d'autres paquets ou modules appropriés (par exemple, vous pouvez vouloir un module principal de haut niveau).

Si vous avez le choix entre la manipulation de dictionnaires ou d'instances de classe, choisissez toujours des instances de classe pour assurer la conformité des modèles MVC.

Pour effectuer le peluchage du code, veuillez utiliser [flake8](#) avec l'option de longueur de ligne maximale fixée à 119. Nous aimerions également voir un rapport généré par [flake8-html](#) dans le repository, sous un répertoire appelé "flake8\_rapport", qui n'affiche aucune erreur et nous assurera que votre code est conforme aux directives PEP 8.

Veuillez utiliser pip pour définir l'environnement Python. Nous aimerions voir un document README.md dans le repository contenant des instructions sur la façon de configurer et d'exécuter le programme, et de générer un nouveau rapport flake8.

## ANNEXE

### LE SYSTÈME SUISSE DES TOURNIS

#### GÉNÉRATION DES PAIRES DE JOUEURS

Nos paires sont générées selon le système de tournoi suisse. Vous trouverez ci-dessous la manière dont nous préférons mettre en œuvre ce système :

1. Au début du premier tour, triez tous les joueurs en fonction de leur **classement**.
2. Divisez les joueurs en deux moitiés, une **supérieure** et une **inférieure**. Le meilleur joueur de la moitié supérieure est jumelé avec le meilleur joueur de la moitié inférieure, et ainsi de suite. Si nous avons huit joueurs triés par rang, alors le joueur 1 est jumelé avec le joueur 5, le joueur 2 est jumelé avec le joueur 6, etc.
3. Au prochain tour, triez tous les joueurs en fonction de leur **nombre total de points**. Si plusieurs joueurs ont le même nombre de points, triez-les en fonction de leur rang.
4. Associez le joueur 1 avec le joueur 2, le joueur 3 avec le joueur 4, et ainsi de suite. **Si le joueur 1 a déjà joué contre le joueur 2**, associez-le plutôt au joueur 3.
5. Répétez les étapes 3 et 4 jusqu'à ce que le tournoi soit terminé.

## ÉQUILIBRER LES COULEURS

Un tirage au sort des joueurs définira qui joue en blanc et qui joue en noir ; il n'est donc pas nécessaire de mettre en place un équilibrage des couleurs.