

A close-up photograph of a chessboard with several dark wooden pieces. In the foreground, a white king piece is lying on its side on a light square. Several dark pieces, including a king, a queen, and several pawns, are standing upright on various squares. The background is blurred, showing warm, out-of-focus lights.

# Chess Tournament P4 OpenClassrooms

---

SHMUEL BITAN



# Player

- *Principales Fonctions :*
  - Stocke les informations d'un joueur d'échecs (nom, prénom, date de naissance, etc.).
- *Utilisation :* Créez et gérez des objets joueur pour les tournois.



# PlayerManager

- Classe "PlayerManager"
- - Principales Fonctions :
  - - Gère la liste des joueurs.
  - - Charge et sauvegarde les données des joueurs depuis/vers un fichier JSON.
  - - Permet de créer, modifier et supprimer des joueurs.
  - - Utilisation : Gérez la liste des joueurs et leurs données.



# Tournement

- Principales Fonctions :
  - - Stocke les informations d'un tournoi (nom, lieu, date, contrôle du temps, etc.).
  - - Gère les joueurs inscrits dans le tournoi.
  - - Utilisation : Créez et gérez des tournois d'échecs.



# TournamentManager

- Principales Fonctions :
- - Gère la liste des tournois.
- - Organise les pairs de joueurs pour les matchs.
- - Enregistre les résultats des matchs et gère les scores des joueurs.
- - Utilisation : Gérez les tournois et les matchs d'échecs.



# Match

- Principales Fonctions :
  - - Représente un match entre deux joueurs.
  - - Permet de saisir le résultat du match (victoire, défaite, match nul).
  - - Gère les scores des joueurs en fonction du résultat.
  - - Utilisation : Enregistrez les résultats des matchs dans les tournois.

# RoundManager

- Principales Fonctions :
  - - Gère l'organisation des rounds dans un tournoi.
  - - Enregistre les résultats des matchs et les sauvegarde dans un fichier.
  - - Gère le déroulement des matchs.
  - - Utilisation : Organise et gère les rounds dans les tournois d'échecs.

---

# Cas d'utilisation : nouveau joueur

On crée une instance `player_manager`

On crée un nouveau joueur grâce à la vue `player`

On sauvegarde le joueur sur le json en tant que dict

On pourra récupérer le joueur grâce à son id lors de la création d'un tournoi pour l'y ajouter



---

# Cas d'utilisation : création de tournoi 1

- On initialise un Tournament\_manager et un player\_manager
- On crée un nouvel objet tournoi grâce à la vue
- On récupère le nombre de joueur et de tour depuis le tournoi
- Tant que le nombre de joueur ajoute est inférieur au nombre de joueur a ajouter on va récupérer un joueur grâce à son id et l ajouter à la liste de joueur



## Cas d'utilisation : creation de tournoi 2

- Tant que le nombre de round jouée est inférieur au nombre de round on relance un nouveau match
- Finalement on sauve le tournoi avec les joueurs sur le json et les rounds sont sauvé au fur et à mesure avec l'id du tournoi pour les récupérer