Controllers/Tournoi\_Controller.py

import json

from models.tournoi\_model import TournoiManager

class TournoiController

def \_\_init\_\_(self):

        self.tournoi\_manager = TournoiManager()

def ajouter\_tournoi(self, nom, date\_debut, date\_fin, nb\_max\_joueurs, nb\_rondes, type\_tournoi)

def modifier\_tournoi(self, index, nom=None, date\_debut=None, date\_fin=None, nb\_max\_joueurs=None, nb\_rondes=None, type\_tournoi=None)

def supprimer\_tournoi(self, index)

def ajouter\_joueurs\_au\_tournoi(self, tournoi\_index, joueurs)

def supprimer\_joueur\_du\_tournoi(self, tournoi\_index, joueur)

def creer\_ronde(self, index\_tournoi)

def modifier\_ronde(self, index\_tournoi, ronde\_numero)

def supprimer\_ronde(self, index\_tournoi, ronde\_numero)

def afficher\_resultats\_ronde(self, index\_tournoi, ronde\_numero)

def afficher\_classement\_final(self, index\_tournoi)

def trouver\_tournoi\_par\_index(self, index):

def appariement\_ronde(self, index\_tournoi)

def obtenir\_resultats\_ronde(self, index\_tournoi)

def appariement\_ronde(self, index\_tournoi)

def obtenir\_classement\_ronde(self, index\_tournoi)

models/tournoi\_mode.py

class Tournoi

def \_\_init\_\_(self, index, nom, date\_debut, date\_fin, nb\_max\_joueurs, nb\_rondes, type\_tournoi, joueurs=None):

        self.index = index

        self.nom = nom

        self.date\_debut = date\_debut

        self.date\_fin = date\_fin

        self.nb\_max\_joueurs = nb\_max\_joueurs

        self.nb\_rondes = nb\_rondes

        self.type\_tournoi = type\_tournoi

        self.joueurs = joueurs if joueurs is not None else []

        self.nombre\_inscrits = len(self.joueurs)  # Ajout de l'attribut nombre\_inscrits

        self.rondes = []  # Liste pour stocker les tours

def generer\_donnees\_json(self)

def creer\_fichier\_json(self)

def creer\_ronde(self)

def ajouter\_ronde(self, ronde)

def mise\_a\_jour\_fichier\_json(self)

def ajouter\_joueur\_au\_tournoi(self, joueurs)

def supprimer\_joueur\_au\_tournoi(self, joueur)

def charger\_joueurs(self)

def sauvegarder\_joueurs

def classement(self)

class Match

class Tour

def \_\_init\_\_(self, numero, date, statut):

self.numero = numero

        self.date = date

        self.statut = statut

        self.matchs = []  # Liste pour stocker les matchs

  self.rondes = []  # Initialisation de l'attribut pour stocker les rondes

self.joueurs: List[Joueur] = []  # Spécifiez le type de joueurs comme une liste de Joueur

def appariement\_tour(self)

class TournoiManager

FICHIER\_JSON = "data/tournaments.json"

def \_\_init\_\_(self):

        self.tournois = []

        self.charger\_tournois()

  def charger\_tournois(self)

def sauvegarder\_tournois(self)

def modifier\_tournoi(self, index, nom=None, date\_debut=None, date\_fin=None, nb\_max\_joueurs=None, nb\_rondes=None, type\_tournoi=None)

def supprimer\_tournoi(self, index)

def ajouter\_joueurs\_au\_tournoi(self, tournoi\_index, joueurs)

def supprimer\_joueur\_du\_tournoi(self, tournoi\_index, joueur)

def convertir\_dict\_vers\_tournoi(self, data)

def charger\_joueurs\_par\_index(self, joueurs\_index)

def convertir\_tournoi\_vers\_dict(self, tournoi)

def trouver\_tournoi\_par\_index(self, index)

views/tournoi\_vue.py

class TournoiVue

def \_\_init\_\_(self):

self.tournoi\_controller = TournoiController()

      self.joueur\_manager = JoueurManager()

def afficher\_menu(self)

def saisir\_tournoi(self)

def valider\_format\_date(date\_str)

def modifier\_tournoi(self)

def supprimer\_tournoi(self)

def afficher\_liste\_tournois(self)

def saisir\_index\_tournoi(self)

def saisir\_joueurs\_participants(self, tournoi\_index)

def afficher\_details\_tournoi(self)

def supprimer\_joueur\_tournoi(self, tournoi\_index)

def creer\_tour(self, index\_tournoi)

def appariement\_tour(self, index\_tournoi)

def afficher\_resultats\_tour(self, index\_tournoi)

def afficher\_classement\_tour(self, index\_tournoi)