

Travail Pratique – Gestionnaire de tournoi de jeux vidéo

Cours : 420-2C3-MA Programmation orientée objet

Professeur : Eva Terriault

Pondération : 10 % de la note finale

Intelligence artificielle : Autorisée selon la politique du département

Date de remise : Lundi 12 mai à 23h59

Mode de remise : Voir la section Remise

Travail individuel ou en équipe de 2

Contexte

Vous êtes mandatés pour développer une application de gestion de tournois de jeux vidéo (un joueur contre un autre).

L'application devra permettre de :

- Créer des tournois
- Charger des listes de joueurs et de matchs
- Saisir les résultats des matchs
- Produire un rapport final

Ce travail évaluera :

- La programmation orientée objet
 - La gestion de fichiers (CSV, JSON, texte)
 - La modélisation UML
 - L'utilisation de Git pour la gestion du projet
-

Travail à faire

Dans ce travail, vous devez :

- Élaborer des diagrammes UML
- Compléter le projet de gestion de tournoi
- Utiliser Git pour le développement et la remise

UML

Vous devez produire **deux diagrammes UML** :

1. Diagramme de classes

- Représentant vos classes (**Tournoi**, **Joueur**, **Match**) et leurs relations
- Présentant les attributs et principales méthodes

2. Diagramme au choix parmi :

- Diagramme de séquence (ex.: déroulement de la saisie d'un score)
- Diagramme d'activité (ex.: processus de création d'un tournoi)
- Diagramme de cas d'utilisation (ex.: actions possibles pour un utilisateur)

Consignes UML

- Les diagrammes doivent être réalisés à l'ordinateur (aucune version papier).
- Les fichiers doivent être remis au format `.png` ou `.jpg`.
- Déposez vos fichiers dans le dossier **UML/** fourni dans le projet.
- Utilisez une nomenclature simple et claire (en français).
- Les diagrammes doivent être propres, lisibles et correspondre au code.

Outils recommandés

- [draw.io \(diagrams.net\)](#) — Très simple et gratuit (navigateur)
 - [Lucidchart](#) — Application en ligne avec plan gratuit limité
 - [PlantUML](#) — Génération de diagrammes à partir de texte
-

Lecture et écriture de fichiers

Vous devrez compléter deux fichiers principaux :

- `tournoi.py` :
 - Charger les joueurs depuis un fichier CSV
 - Charger les matchs depuis un fichier CSV
 - Saisir les scores des matchs (déjà implémenté)
 - Afficher le classement final (déjà implémenté)
 - Sauvegarder le tournoi en JSON
 - Générer un rapport final en texte
- `utils.py` :
 - Lire un fichier CSV (`lire_csv`)
 - Sauvegarder un fichier JSON (`sauvegarder_json`)
 - Écrire du texte dans un fichier texte (`ecrire_texte`)

Vous devez respecter les signatures de méthodes et suivre précisément les explications données dans les commentaires.

Important: Le fichier `utils.py` ne devrait pas contenir de la manipulation des objets (Joueur, Match ou Tournoi). Il ne devrait être en charge que de lire ou écrire selon les arguments passés dans les méthodes. La manipulation des objets devrait se faire dans le fichier `tournoi.py`.

README

Vous devez remplir le fichier `README.md` en indiquant :

- Le nom et le matricule des membres de l'équipe
- (Optionnel) Des instructions supplémentaires pour exécuter votre projet

Note : Le format habituel d'un README est en **Markdown**. Bien que le format Markdown ne soit pas évalué dans ce travail, son apprentissage est recommandé pour votre futur développement professionnel.

Utilisation de Git

Vous devez utiliser Git pour versionner votre code tout au long du développement :

- Faire **des commits réguliers et significatifs**
- Utiliser **des branches** pour le développement
- **Fusionner** vers la branche **main** avant la remise

Exigences Git

- Votre dépôt doit contenir un minimum de **5 commits significatifs**.
 - Vous devez effectuer au moins **un merge** entre une branche de développement et **main**.
 - Votre dépôt GitHub ou GitLab doit être **public**.
-

Fichiers fournis

Vous allez baser votre travail sur le dépôt git suivant : <https://github.com/eterriault-prof/2025H-420-2C3-TP2>

Arborescence du projet :

```
projet_tournoi/  
├── UML/ (dossier vide pour vos diagrammes UML)  
├── data/  
│   ├── joueurs.csv (liste de joueurs)  
│   └── matchs.csv (liste de matchs entre joueurs)  
├── main.py  
├── tournoi.py  
├── joueur.py  
├── match.py  
├── utils.py  
└── README.md (vide à compléter par vous)
```

Explications des fichiers fournis

- **main.py** : Fichier principal qui contient le menu de l'application. Vous n'avez pas à le modifier. Il appelle les méthodes à compléter dans **tournoi.py** et **utils.py**.
- **joueur.py** : Fichier contenant la classe **Joueur** (pseudo, nombre de victoires). Ce fichier est complet.
- **match.py** : Fichier contenant la classe **Match** (deux joueurs, scores). Ce fichier est complet.
- **tournoi.py** : À compléter. Classe **Tournoi** pour gérer les joueurs, matchs, résultats, classement, sauvegarde et rapport.

- **utils.py** : À compléter. Fonctions pour lire des CSV, sauvegarder des fichiers JSON et écrire des fichiers texte.
 - **data/** : Dossier contenant des jeux de données (**joueurs.csv** et **matches.csv**) pour tester votre application.
 - **UML/** : Dossier où déposer vos diagrammes UML remis en format **.png** ou **.jpg**.
 - **Documents/** : Dossier contenant l'énoncé du TP et le gabarit de la grille de correction.
 - **README.md** : Fichier à remplir avec les informations de votre équipe et, si désiré, des instructions supplémentaires.
-

Remise

Vous devez remettre sur Omnivox **seulement** un fichier texte contenant un lien vers votre dépôt Git

Le dépôt doit :

- Rester en ligne au moins jusqu'à la fin de la session
- Contenir votre code fonctionnel sur la branche **main**
- Ne pas contenir de nouveaux commits après la date de remise

Le travail corrigé sera celui qui se trouve sur la branche **main** au moment de la date limite.

Grille d'évaluation

La grille d'évaluation se trouve dans le dossier **documents** du dépôt git fourni.

Note : La note finale sera **individuelle**.

Elle peut être **ajustée à la hausse ou à la baisse** selon votre participation au projet (évaluée grâce à vos commits Git et aux observations du professeur en classe).
