# Projet POO3

Le but de ce petit projet est de valider l'ensemble des connaissances vues en POO3.

Pour ce faire, vous allez travailler par groupe de 4 (ou 3).

Le projet est de faire un système de gestion de personnages et d'armes pour des jeux de rôles.

Il doit y avoir une page principale (accueil) et 2 **onglets** avec le CRUD des personnages et armes.

Une api va gérer les combats.

## Base de données

- Un fichier JSON va contenir la configuration
- La base de données est SQLite.

  Connect To The SQLite Database Using SQLite JDBC Driver (sqlitetutorial.net)

  La base se trouvera dans : `C:/sqlite/db/poo3.db`

## Interface IHM

- En java FX
- Au chargement il y a deux boutons : Gestion Personnages et Gestion Armes.
- Il y a une indication du nombre de personnages dans la db et des armes.

## Pour les CRUD

- Afficher la liste des Armes ou Personnages
- Pour chaque item on doit avoir : voir, modifier ou effacer
- Avoir un bouton ajouter
- Ouvrir une nouvelle fenêtre pour les actions demandant de voir la fiche de l'arme ou du personnage.
- Pour le Delete, demander une validation.

## Attributs du personnage

- ID (Auto)
- Nom
- Point de vie (PV) (max 1000)
- Manna (max 100)

## Attributs d'arme

- ID (Auto)
- Nom
- Dégats (Max 100)

## API

Créer les API pour les attaques et ce qui est nécessaire autour.

- /personnage : renvoie la liste des personnages (id et nom)
- /personnage/{id} : renvoie les détails d'un personnage
- /arme : renvoie la liste des armes (id et nom)
- /arme/{id} : renvoie les détails d'une arme
- /attaque/{id personnage}/{id arme}

  Renvoie les détails du personnage attaqué.

NB : les PV ne descendent pas sous zéro !

## Git Flow

- On utilise GitFlow
- Faire des micro commits
- La branche master/main est celle que nous corrigerons
- La branche develop va consolider les développements des branches de features par élève.
- **Si inexistant ou incomplet -> Ajourné !**

## JavaDoc obligatoire

**Si inexistante ou incomplète -> Ajourné !**

## Journal de bord

- Il devra comprendre pour chaque semaine jusqu'à la restitution ce qui a été fait et par qui.
- Il sera stocké dans un dossier carnet de bord dans le GitHub et géré par le profil 1
- **Si inexistante ou incomplète -> Ajourné !**

## Découpage des tâches

### Profil 1

- Gérer le gitHub et les pull requests
- Faire la base de l'IHM
- Faire le Journal de bord et suivre le projet (dans un folder JournalDeBord).
- Faire la partie "lecture du JSON de config" (réfléchir au format)
- Faire l'UML
- Ecrire la doc JavaDoc de sa partie

### Profil 2

- Faire le CRUD sur les personnages
- Utiliser l'IHM du profil 1
- Utiliser la gestion du fichier de config JSON du profil 1
- Ecrire la doc JavaDoc de sa partie
- Renvoyer sa partie pour le journal de bord de la semaine.

Profil 3

- Faire le CRUD sur les armes
- Utiliser l'IHM du profil 1
- Utiliser la gestion du fichier de config JSON du profil 1
- Ecrire la doc JavaDoc de sa partie
- Renvoyer sa partie pour le journal de bord de la semaine.

Profil 4

- Faire les API's
- Utiliser la gestion du fichier de config JSON du profil 1
- Ecrire la doc JavaDoc de sa partie
- Renvoyer sa partie pour le journal de bord de la semaine.

**Tous les profils doivent réaliser des tests !**

**(Réfléchir à intégrer le plus possible le travail des autres)**

# Date Limite 24 mai 2024 à 23h59