

# Énoncé — Analyse Programmation

## 2022-2023

### Introduction

Le but de cet examen d'Analyse Programmation est de créer un petit jeu de combat où plusieurs personnages provenant de deux équipes s'affrontent dans une arène. Le jeu en langage Python doit s'exécuter dans une console en ligne de commande loggant toutes les actions effectuées par le programme.

Les combats sont composés de deux équipes de 10 personnages de différentes classes (voir plus loin dans l'énoncé). Ces personnages, choisis au hasard à chaque début de partie, sont issus d'une base de données créée par vos soins. La partie se termine une fois que l'équipe adverse n'a plus de personnages vivants.

Ce travail est à réaliser de façon individuelle.

### Déroulement de la partie

À chaque lancement de l'application et donc du jeu, 20 personnages sont créés au hasard et enregistrés dans la base de données. Ensuite, parmi ces nouveaux personnages et les personnages vivants restants (de la bd), 20 sont choisis aléatoirement pour composer les équipes. À la fin de la partie, les points de vie restants des personnages sont enregistrés en base de données. Les personnages ayant 0 point de vie sont aussi enregistrés, mais ces joueurs ne pourront plus être sélectionnables pour faire d'autres combats vu qu'ils sont éliminés. Un personnage peut être affecté à plusieurs équipes, mais pas pour le même combat.

Dans un but futur de faire des statistiques après chaque partie, le résultat du combat ainsi que les tactiques utilisées (voir plus loin dans l'énoncé), le nombre de personnages encore en vie et la liste des personnages ayant participé aux combats doivent être enregistrés en base de données.

Chaque personnage correspond à un Thread.

La partie ne se déroule pas en tour par tour. Chaque personnage attend (1000 / valeur d'initiative) millisecondes, puis effectue une action (écrire l'algorithme de choix), puis attend (1000 / valeur d'initiative), puis effectue une action, etc.

Un personnage possède plusieurs caractéristiques le définissant, ces caractéristiques sont définies par sa classe, il en existe en tout 4 :

- Le guerrier
  - Possède une valeur d'attaque comprise entre 70 et 90
  - Possède une valeur de défense comprise entre 70 et 90
  - Possède une valeur de points de vie comprise entre 120 et 150
  - Possède une chance de coup critique comprise entre 5 % et 7 %
  - Possède une chance de parade comprise entre 40 % et 60 %
  - Possède une initiative comprise entre 40 et 60
- Le voleur

- Possède une valeur d'attaque comprise entre 40 et 60
- Possède une valeur de défense comprise entre 30 et 50
- Possède une valeur de points de vie comprise entre 70 et 80
- Possède une chance de coup critique comprise entre 15 % et 20 %
- Possède une chance d'esquive comprise entre 40 % et 70 %
- Possède une initiative comprise entre 75 et 90
- Le mage
  - Possède une valeur d'attaque comprise entre 100 et 150
  - Possède une valeur de défense comprise entre 20 et 40
  - Possède une valeur de points de vie comprise entre 60 et 70
  - Possède une chance de coup critique comprise entre 5 % et 7 %
  - Possède une initiative comprise entre 60 et 70
- Le prêtre
  - Choisit, soit de soigner, soit d'attaquer, à chacun de ses tours
    - Il soigne un de ses alliés dont la vie n'est pas à son maximum
    - Il soigne à hauteur de sa propre défense / 4
  - Possède une valeur d'attaque comprise entre 30 et 60
  - Possède une valeur de défense comprise entre 60 et 80
  - Possède une valeur de points de vie comprise entre 70 et 90
  - Possède une chance de coup critique comprise entre 5 % et 7 %
  - Possède une chance de parade comprise entre 30 % et 50 %
  - Possède une initiative comprise entre 50 et 60

Un coup critique ignore la défense. Les classes sont attribuées de façon aléatoire tout comme les ses caractéristiques.

Vous devez afficher un message à chaque action d'un personnage, décrivant l'action, les points de dégâts, etc. Chaque personnage possède aussi un **nom et un prénom**, qui sera affiché avec chaque action. Je vous conseille de vous baser sur un dictionnaire de nom pour les attribuer.

Le coup critique n'est calculé qu'après les manœuvres de défense. Si un personnage esquive ou pare une attaque, aucun coup critique ne peut être effectué pour cette attaque. Une esquive ou une parade permettent au défenseur de ne pas prendre de dégâts. Les dégâts ne sont calculés qu'après calcul des manœuvres défensives et du coup critique.

Le nombre de dégâts est égal à valeur de l'attaque de l'attaquant à laquelle on soustrait la valeur de défense du défenseur.

Il vous est demandé de créer plusieurs tactiques (algorithmes) pour les combats. À vous de créer ces algorithmes. Voici deux exemples de tactiques de combat :

- Viser les prêtres en premier
- Attaquer au hasard

## Modalités

Il vous est demandé d'utiliser les règles de l'orienté objet pour cette évaluation ! La bonne utilisation des concepts (classes, héritages, threads...) sera évaluée. Une mauvaise utilisation de ces concepts entraînera un échec.

Cet énoncé doit être terminé au plus tard le jour précédent l'examen. En plus du programme, vous devez aussi réaliser :

- Les diagrammes d'analyse adéquats (à vous de juger, votre note dépendra aussi de vos choix)
- Une description textuelle des algorithmes (tactique de combat) accompagné de leurs diagrammes
- Une base de donnée (format SQL)
- L'analyse de la base de donnée

Tout le code doit se trouver sur le GitLab ainsi que les différents documents écrits/diagrammes le jour avant l'examen.

Ce travail sera évalué lors d'une épreuve orale le jour de l'examen.