

## Projet JAVA POO

Dans ce projet, nous allons mettre en place les bases d'un jeu de plateau de type **HearthStone**.

Pour rappel, il s'agit d'un jeu de plateau où 2 champions s'affrontent.

Ces champions sont caractérisés par un id, un nom, un nombre de point de vie et une capacité spéciale. Un champion ne peut pas attaquer sauf si sa capacité spéciale lui permet.

Chaque champion possèdera un ensemble de cartes qu'ils vont pouvoir poser sur le plateau chacun leur tour.

Chaque carte invoquera un monstre sur le plateau, un monstre est caractérisé par un id, un nom, un nombre de point de vie et une force d'attaque.

Nous aurons plusieurs types de monstre :

- Des « classiques » qui attaquent l'ennemi.
- Des protecteurs qui empêchent les monstres adverses d'attaquer le champion directement ou d'autres monstres non protecteurs.
- Des soigneurs qui ne peuvent attaquer mais peuvent soigner un allié ou un ennemi
- Des mascottes, qui boostent une cible alliée ou ennemie
- Autre en fonction de votre imagination (mort vivant qui se réanime une fois tué, pilote de ligne qui s'enfuit quand il est invoqué, etc...)

Chaque type de monstre aura sa propre manière de jouer son tour de jeu.

Déroulement d'un tour de jeu :

- Le champion peut décider de poser une carte
- Le champion décide d'utiliser sa capacité spéciale ou non.
- Le champion et chacune de ses cartes peuvent attaquer une cible
- Le tour se fini et le tour du champion adverse se lance jusqu'à ce qu'un des champions est KO (ses points de vie  $\leq 0$ ).

Ce qui est demandé techniquement :

- Utilisation d'au moins 1 interface
- Mise en place d'un héritage minimum
- Tracer les informations importantes dans un fichier de log dans le format que vous souhaitez (ex : « champion 1 invoque protecteur 1 », « monstre 3 tue le champion ennemi, partie terminée », etc...)
- Faire minimum 5 tests unitaires, les 2 suivants sont obligatoires :
  - Simuler l'attaque d'un monstre sur un autre avec des points d'attaque égaux aux points de vie du monstre attaqué. Vérifier que ce dernier ait bien 0 points de vie et disparaît du plateau.
  - Créez un test qui instancie un combat entre 2 Champions, un des deux met KO l'autre. Vérifiez que le combat s'arrête avec le bon vainqueur.

- Faites 3 autres tests sur les fonctionnalités principales de votre application ou sur ce qu'il vous semble important, les TU sur les getters et setters ne sont pas toujours très utiles...
- Respecter les conventions de codage en Java (nom de classe en UpperCamelCase, nom de variable et de méthode en lowerCamelCase, des bonnes indentations, etc...)
- Insister sur les commentaires et/ou la javadoc
- (Optionnel) Utilisation des stream et des fonctions anonymes.

Date de rendu : 25 Mars 2023

**Attention !!** La première ligne du Main doit écrire les noms et prénoms du binôme dans la console