

DungeonMaster

Afin de mettre en pratique l'ensemble des éléments vus cette année, je vous propose de réaliser un jeu-vidéo de type RPG. Dans ce jeu, vous incarnez une héroïne devant affronter 3 types de monstres différents afin d'aller le plus loin possible dans le donjon.

Ce projet va être découpé en 3 parties afin de pouvoir avoir un rendu final fonctionnel. Vous pouvez le réaliser seul ou en groupe de 2 maximum. Le but est de mettre en pratique les connaissances sur la POO vue en cours. Vous pouvez vous aider des cours, des corrections mais évitez d'utiliser des technologies trouvable sur le net que nous n'avons pas vu ensemble.

Première partie :

Dans le contexte de la POO, nous allons créer des classes permettant la génération des monstres et de notre héros. Dans le jeu, chaque étage du donjon est composé de 10 salles. Pour chaque étage, vous avez le même type de monstre avec un lieutenant salle 5 et un boss salle 10.

Je vous propose donc de créer 2 classes différentes :

1- Le héro

Cette classe contiendra l'ensemble des caractéristiques de votre héros avec les propriétés et les méthodes suivantes (Accesseurs et Mutateurs non comptés) :

- Nom
- Prénom
- Force
- Dégâts
- Défense
- Point de vie
- Sac de potions
- Sac d'armes
- Méthode d'attaque prenant en paramètre la cible et qui renvoie true sur la cible est vaincue sinon qui mets à jour les points de vie de la cible (défense à prendre en compte)
- Méthode qui permet de récupérer automatiquement de l'équipement. Elle prend en paramètre le nom de l'équipement et doit vérifier si le nombre de potions (10) et d'équipement (4) maximum n'est pas atteint. Elle doit aussi automatiquement ajouter dans le sac correspondant du personnage l'objet dans le cas d'une arme ajouter 4 aux dégâts de base du personnage
- Méthode d'utilisation de la potion : elle doit permettre de retrancher une potion et d'augmenter la vie du personnage de 10 sans dépasser le nombre de point de vie maximum

Afin de garantir une part d'aléatoire, lors de l'initialisation de mon personnage la force (3 – 10) et la défense (1 – 5) sont générées de manière aléatoire. Les dégâts provoqués sont égaux au produit de la force par les dégâts de base (3). Ces éléments étant générés durant l'initialisation, je dois donc prévoir une méthode init qui sera appelé dans mon constructeur.

Pour simplifier la gestion, je vous encourage à utiliser des variables statiques vous permettant de sauvegarder au sein de votre classe les éléments qui serviront vos calculs lors de l'initialisation. On peut ainsi imaginer les propriétés static suivantes (pour les utilisateur `nomDeLaClasse.VariableStatic`) :

- Dégâts = 3
- MinForce = 3
- MaxForce = 10
- MinDef = 1
- MaxDef = 5

2- Monstres

Je vous invite ensuite à créer une classe qui regroupera l'ensemble des méthodes et propriétés communes à toutes les créatures. Elle sera composée des propriétés et des méthodes suivantes (sans compter Accesseur et Mutateur) :

- Nom
- Attaque
- Défense
- Point de vie
- Images
- Target
- Méthode d'attaque qui prend en paramètre la cible, qui mets à jour les points de vie de la cible en diminuant l'attaque du monstre par la défense de la cible. Elle renvoie true si sa cible est morte
- Méthode d'ajout dans le dom : vide et construite dans les classes filles (partie 2)
- Méthode de suppression du dom : vide et construite dans les classes filles (partie 2)
- Méthode permettant de lâcher une potion / arme : vide et construite dans les classes filles (partie2)
- Aucune démarche particulière n'est à réaliser lors de l'initialisation (constructor)

Barème

Héros

Consignes	Notes
Structure de base de la classe	/5
Accesseur et mutateur	/8
Méthode d'attaque	/3
Méthode de récupération du « loot »	/5
Méthode d'utilisation de la potion	/4
Présence de valeur static	/2
Méthode d'initialisation	/5
Classe fonctionnelle	/2
Affichage des caractéristiques du héros dans la console	/2

Monstres

Consignes	Notes
Structure de base de la classe	/5
Accesseur et mutateur	/8
Création des méthodes	/3
Classe fonctionnelle	/2
Affichage des caractéristique du monstre dans la console	/2

La note totale est de 53 et sera ramené à 20. Les groupes constitués seront indiqués avec le lien vers le dépôt ou le dossier du projet accessible dans le drive à votre disposition.

Ce barème est donné à titre indicatif, il peut évoluer selon le niveau moyen de la classe.

Le détail que je vous ai fourni pour la construction des classes ou des fonctions / méthodes peut varier selon les solutions que vous allez trouver. Certaines propriétés peuvent apparaitre pour faciliter la gestion du DOM dans vos classes par exemple. Si c'est le cas, et qu'une telle propriété n'est utilisé qu'au sein de la classe, vous n'êtes pas obligé de créer les accesseurs et mutateurs correspondant. Attention les variables statiques n'ont pas de getters / setters.