TD 1 : jeu classe et objets

Vous connaissez les jeux où nous affrontons une liste de monstre au tour par tour pour réussir une quête ?

Nous allons essayer de faire une base simple de ce type de jeu.

1. Partie 1 : le cœur du jeu

La partie centrale du jeu comprendra le personnage de notre héros, la définition des monstres que notre héros devra affronter ainsi qu'un programme principal (le main) qui mettra en place le jeu.

Nous aurons donc besoin de 2 classes proches l'une de l'autre :

La classe Personnage

Nous considérerons pour le moment qu'un personnage a des points de vie, inflige des dégâts fixes et qu'il a un pourcentage d'armure qui attenue d'autant les dégâts subis.

Personnage
- pv : int - dégâts : double - armure : double
+ Personnage(pv: int, dégâts : doule, armure : double) + Personnage() + attaquer(m : Monstre) + parer() + jouer() + toString() : String

- 2 constructeurs permettant de créer un personnage, le constructeur par défaut (sans paramètre permettra de créer un personnage avec des valeur prédéfinies (ex: 30pv 20 dégâts 0 armure).
- Une méthode **attaquer**, qui permet d'attaquer un monstre passé en paramètre et de lui infliger les dégâts du personnage.
- Une méthode **parer**, qui permet de se mettre en défense pour le prochain tour (ceci peut se faire en augmentant pour 1 tour la valeur de l'armure ex : 50%)
- Une méthode **jouer**, qui permet de choisir l'action (attaquer ou parer).
- Une méthode **toString** qui permettra d'afficher les caractéristiques d'un personnage (pv dégâts armure)
- Ainsi que les accesseurs (ceux-ci pourront avoir une fonction plus importante que le simple accès aux attributs)

Aflokkat – Lieu dit Effrico 20167 Sarrola Carcopino - www.aflokkat.com - contact@aflokkat.com

La classe Monstre

Nous considérerons pour le moment qu'un personnage a des points de vie, inflige des dégâts fixes et qu'il a un pourcentage d'armure qui attenue d'autant les dégâts subis.

Monstre
- pv : int - dégâts : double - armure : double - life : boolean
+ Monstre(pv: int, dégâts : doule, armure : double) + Monstre() + attaquer(m : Monstre) + parer() + jouer() + toString() : String

- 2 constructeurs permettant de créer un monstre, le constructeur par défaut (sans paramètre permettra de créer un personnage avec des valeur prédéfinies (ex: 30pv 20 dégâts 0 armure).
- Une méthode **attaquer**, qui permet d'attaquer un personnage passé en paramètre et de lui infliger les dégâts du monstre.
- Une méthode **parer**, qui permet de se mettre en défense pour le prochain tour (ceci peut se faire en augmentant pour 1 tour la valeur de l'armure ex: 50%)
- Une méthode **jouer**, qui permet de choisir l'action (attaquer ou parer) par exemple en lançant un dé 50% parade 50% attaque pour cela pensez a regarder le fonctionnement de la bibliothèque java.util.Random.
- Une méthode toString qui permettra d'afficher les caractéristiques d'un personnage (pv dégâts armure)
- Ainsi que les accesseurs (ceux-ci pourront avoir une fonction plus importante que le simple accès aux attributs)

  Pensez à une façon de tenir compte de l'armure en infligeant des dégâts !

Un programme principal qui permettra de jouer.

Plusieurs options s'offrent à nous, je vous propose de créer une liste de monstres (pour cela je vous invitent a regarder le fonctionnement de la bibiothèque java.util.arrayList) qui contiendra les monstres que notre héro devra vaincre pour gagner la partie. Ensuite bien sûr de créer un personnage.

La règle du jeu en soit est simple, tant que le héros est en vie et que la liste de monstre n'est pas vide le combat continue. Notre héros affronte le 1er monstre de la liste, si un monstre est vaincu, il disparait de la liste et le suivant prend le relais.

Si la liste se vide, le héros gagne la partie, si notre héro meurt, il perd.

Je vous laisse penser à un affichage pas trop moche dans la console.

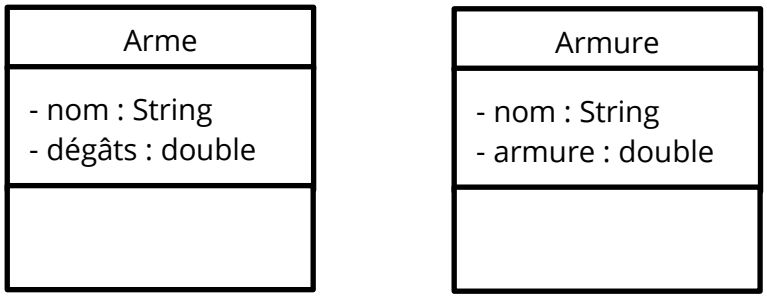
2. Partie 2 : Ajout d'équipement et de butin

Bien maintenant que le cœur de notre jeu fonctionne, pourquoi ne pas ajouter des petites fonctionnalités. Par exemple, pourquoi ne pas ajouter la présence d'équipement. Nous pourrions ajouter des armes et des armures que notre héros pourrait ramasser sur les monstres et porter

Pour cela nous aurons besoin d'ajouter quelques attributs à nos Personnages (Armes et Armures qu'il équiperait et qui amélioreraient les stats) et a nos monstres (Arme ou Armure qui serait remise au personnage qui l'a vaincu à la mort du montre)

Ainsi que 2 classes d'objets :

Arme et Armure avec 2 attributs, un nom et une statistique. Sans comportement propre mais bien sûr avec des constructeurs.



Il va nous falloir adapter les méthodes des classes des monstres pour permettre d'avoir du butin et les méthodes de la classe personnage pour prendre en compte les nouvelles statistiques dépendantes de équipements.

3. Partie 3 : succès échecs et rareté du butin

La dernière partie que je vous propose dans ce TD est d'ajouter une dose de chance dans notre jeu au travers d'un lancé de dé (je vous propose un dé 100 comme en RPG papier)

D'une part je vous propose d'ajouter la notion de rareté et de qualité de butin, un lancé de dé a la mort d'un monstre pourrait permettre de décider s'il y a un butin et si celui-ci est normal, rare ou épique (40% rien, 30% normal, 20% rare, 10% épique) bien sur la rareté joue sur les statistiques de l'équipement.

D'autre part je vous propose aussi d'ajouter les chances de réussite d'un coup (attaque ou parade). Cela implique une nouvelle statistique pour nos personnages et monstre, la chance de réussite. Quand un personnage ou un monstre fait une action on lance un dé qui déterminera si l'action est réussie ou non.

Comme dans un RPG papier je vous propose la règle suivante :

- [0 – 4] : réussite critique (dégâts ou parade X2)
- [4 – chance de réussite] : réussite (dégât ou parade normale)
-] chance de réussite – 94] : échec (dégâts ou parade /2)
- [95 – 99] : échec critique (dégâts ou parade 0)