

II.1102 – Algorithmique et Programmation

TP noté 3

Patrick Wang

Vendredi 17 Janvier 2020

Table des matières

1	Consignes générales	1
2	Modélisation d'un RPG	2
2.1	Description des éléments du jeu	2
2.2	Description des sorts	2
2.3	Description des objets	2
2.3.1	Description des consommables	2
2.3.2	Description des équipements	3
2.4	Questions	3
2.5	Description des actions du joueur	3
2.5.1	Création des classes nécessaires au programme	3
2.5.2	Création d'un personnage	4
2.5.3	Menu du jeu	4
2.5.4	Récupération d'objets	5
2.5.5	Mettre/retirer un équipement	5
2.5.6	Utiliser un objet consommable	6
2.5.7	Jeter un sort	6
2.5.8	Quitter le jeu	6

1 Consignes générales

- Aucun document n'est autorisé. Vous pouvez cependant vous aider de la documentation officielle de Java.
- Prenez garde à bien lire l'énoncé dans son ensemble avant de démarrer les exercices.
- Ce TP est à réaliser **individuellement**.
- Prenez bien garde à bien nommer votre projet en précisant votre groupe d'APP, prénom et nom : `TP3_GX_PRENOM_NOM`.
- Pour ce TP, vous aurez besoin de créer plusieurs classes. Merci de respecter les conventions de nommage pour ces classes.
- Pour déposer votre projet, créez une archive au format ZIP de l'ensemble de votre projet. Les formats RAR ne seront pas acceptés.
- Une boîte de dépôt est créée sur Moodle afin de déposer vos projets. Celle-ci se fermera 15min après la fin du temps réglementaire.
- Toute tentative de triche sera sanctionnée d'un 0.

- Un point sur 20 est attribué à la lisibilité de votre code. Cela concerne l'indentation de votre code, le choix des noms de variables, et la présence de commentaires pertinents.

2 Modélisation d'un RPG

2.1 Description des éléments du jeu

Un *Role Playing Game* est un jeu dans lequel le joueur incarne un personnage évoluant dans un univers défini. Dans le RPG que nous souhaitons concevoir, trois classes de personnages sont jouables : le Guerrier, l'Archère, et le Sorcier. Chaque personnage est caractérisé par :

- Un nom (chaîne de caractères) ;
- Des caractéristiques (Force, Magie, Dextérité, Vitalité) qui prennent des valeurs entières ;
- Des points de vie (actuels et maximum) ;
- Des points de mana (actuels et maximum).

Les personnages peuvent tous faire appel à sorts lors de leurs combats. Un personnage peut gagner l'accès à un sort grâce à un *parchemin de sort*, dont nous détaillerons les caractéristiques en Section 2.2. Les sorts sont quant à eux caractérisés par :

- Un nom (chaîne de caractères) ;
- Un nombre de dégâts infligés (entiers) ;
- Un nombre de points de mana pour lancer le sort.

Enfin, les personnages peuvent obtenir des objets durant leurs aventures. Ces objets se décomposent en deux types :

- Les équipements ;
- Les consommables.

La description de ces objets est donnée en Section 2.3.

2.2 Description des sorts

Pour simplifier, il n'y a que trois sorts dans le jeu :

- Boule de feu, qui coûte 25 points de mana et qui inflige 30 points de dégâts ;
- Éclair en série, qui coûte 5 points de mana et qui inflige 8 points de dégâts ;
- Blizzard, qui coûte 15 points de mana et qui inflige 12 points de dégâts.

Pour pouvoir lancer l'un de ces sorts, il faut que le personnage ait utilisé un parchemin de sort correspondant. Il y a donc trois parchemins différents, un pour chaque sort. Une fois qu'un sort est appris, il n'est pas nécessaire de le réapprendre et le personnage peut lancer ce sort autant de fois qu'il le souhaite.

2.3 Description des objets

Comme mentionné précédemment, les objets se décomposent en équipements et en consommables. Une description de chacun de ces deux types d'objets est donnée ci-après.

2.3.1 Description des consommables

Il existe trois types de consommables :

- Les potions de vie qui rendent jusque 30 points de vie ;
- Les potions de mana qui rendent jusque 30 points de mana ;
- Les parchemins de sorts qui permettent à un personnage d'apprendre le sort en question. Puisqu'il y a trois sorts différents, il y a trois parchemins différents.

Ces objets étant consommables, l'utilisation par un joueur de l'un de ces objets conduit à sa destruction. On note aussi que l'utilisation de l'un de ces objets va sûrement être implémenté par une méthode spécifique.

2.3.2 Description des équipements

Il existe trois emplacements pour les équipements :

- Les armes ;
- Les boucliers ;
- Les armures.

Les armes sont caractérisées par un nom (chaîne de caractères) et un nombre de dégâts infligés (entiers). On distingue deux types d'armes : les épées et les haches. Un exemple d'objet peut être le suivant :

Épée

Nom : Irebleue
Dégâts : 6

Les boucliers sont caractérisés par un nom (chaîne de caractères) et une chance de bloquer une attaque (réel inférieur strictement à 1). Un exemple d'objet peut être le suivant :

Bouclier

Nom : Bouclier Tempête
Chance de bloquer : 0.5

Les armures sont caractérisées par un nom (chaîne de caractères) et un nombre d'armure. Un exemple d'objet peut être le suivant :

Armure

Nom : Bravoure d'Arkaine
Armure : 50

2.4 Questions

2.5 Description des actions du joueur

Dans ce jeu, le joueur pourra :

- Créer un nouveau personnage ;
- Récupérer des objets ;
- Mettre un équipement ;
- Retirer un équipement ;
- Utiliser un objet consommable ;
- Jeter un sort.

2.5.1 Création des classes nécessaires au programme

En suivant la description du RPG donnée en Section 2, créez les classes dont vous aurez besoin pour réaliser les actions décrites plus haut. N'oubliez pas de déclarer les attributs et méthodes utiles au programme.

En plus de ces classes, vous devrez aussi créer une classe **Main** et une classe **Game**. La classe **Main** va simplement instancier un objet de type **Game**, puis faire appel à une méthode **start()**.

L'extrait de code suivant vous indique comment compléter la classe **Main**.

```
public class Main {  
    public static void main(String[] args) {  
        Game game = new Game();  
        game.start();  
    }  
}
```

La classe `Game` va, quant à elle, permettre au joueur de réaliser les actions décrites en Section 2.5. Ces actions pourront être implémentées par des méthodes dédiées, et vous pourrez aussi déclarer les attributs de votre choix.

2.5.2 Création d'un personnage

Au démarrage du jeu (c'est-à-dire, lorsque la méthode `start()` est appelée), on va demander au joueur de créer un nouveau personnage. Pour cela, on va lui demander de choisir entre créer un guerrier, une archère, ou un sorcier. Puis l'utilisateur devra lui donner un nom.

Exemple d'interaction (avec des saisies du joueur comme illustration) :

Indiquez la classe que vous voulez jouer :

```
(1) Guerrier  
(2) Archère  
(3) Sorcier  
3
```

Vous souhaitez créer un Sorcier ! Quel nom souhaitez-vous lui donner ?

Tal Rasha

2.5.3 Menu du jeu

Une fois le personnage créé, un menu va permettre au joueur d'effectuer un certain nombre d'actions. Ce menu pourra se présenter sous la forme suivante :

Faites votre choix :

```
(1) Récupérer un objet  
(2) Mettre un équipement  
(3) Retirer un équipement  
(4) Utiliser un objet consommable  
(5) Jeter un sort  
(6) Quitter le jeu
```

Complétez la méthode `start()` pour que le menu suivant soit affiché après la création du personnage :

1. Afficher le menu ;
2. Recueillir la saisie du joueur ;
3. Effectuer l'action voulue ;
4. Répéter les étapes 1 – 3 tant que le joueur ne quitte pas le jeu.

2.5.4 Récupération d'objets

Dans la classe `Game`, implémentez une méthode qui permet au joueur de récupérer un objet dans son inventaire. Il faudra faire la distinction entre équipement et objet consommable.

Par exemple, pour une arme, on pourra avoir les interactions suivantes :

```
Quel type d'objet souhaitez-vous ajouter à votre inventaire ?
(1) Équipement
(2) Consommable
1
Quel type d'équipement souhaitez-vous ajouter à votre inventaire ?
(1) Arme
(2) Bouclier
(3) Armure
1
Nom de l'arme ?
Irebleue
Dégâts de l'arme ?
6
Arme ajoutée à l'inventaire !
```

Par exemple, pour un objet consommable, on pourra avoir les interactions suivantes :

```
Quel type d'objet souhaitez-vous ajouter à votre inventaire ?
(1) Équipement
(2) Consommable
2
Quel type de consommable souhaitez-vous ajouter à votre inventaire ?
(1) Potion de vie
(2) Potion de mana
(3) Parchemin de sort
3
Quel parchemin de sort souhaitez-vous ajouter à votre inventaire ?
(1) Boule de feu
(2) Éclair en série
(3) Blizzard
2
Un parchemin de sort pour Éclair en série a été ajouté à votre inventaire !
```

2.5.5 Mettre/retirer un équipement

Dans la classe `Game`, implémentez une méthode qui permet au joueur de mettre un équipement. Pour cela, on pourra afficher la liste des équipements dans l'inventaire (attention, l'inventaire peut être vide!). Par exemple, on pourra avoir les interactions suivantes :

```
Quel équipement souhaitez-vous porter ?
(1) Irebleue
(2) Dévoreuse de Messerschmidt
(3) Bravoure d'Arkaine
2
Vous avez équipé Dévoreuse de Messerschmidt !
```

Toujours dans la classe `Game`, implémentez une méthode qui permet au joueur de retirer un équipement (attention, un joueur peut avoir aucun objet d'équipé!). Cet équipement sera ainsi renvoyé dans l'inventaire. Par exemple, on pourra avoir les interactions suivantes :

```
Quel équipement souhaitez-vous retirer ?
```

```
(1) Dévoreuse de Messerschmidt
```

```
(2) Bouclier Tempête
```

```
(3) Bravoure d'Arkaine
```

```
2
```

```
Vous avez retiré Bouclier Tempête !
```

2.5.6 Utiliser un objet consommable

Dans la classe `Game`, implémentez une méthode qui permet au joueur d'utiliser un objet consommable. Pour cela, on pourra afficher la liste des objets consommables dans l'inventaire (attention, l'inventaire peut être vide!). Par exemple, on pourra avoir les interactions suivantes :

```
Quel objet consommable souhaitez-vous utiliser ?
```

```
(1) Potion de vie
```

```
(2) Potion de vie
```

```
(3) Potion de mana
```

```
(4) Parchemin de sort : Éclair en série
```

```
2
```

```
Vous avez bu une potion de vie !
```

Attention, un objet consommé est retiré de l'inventaire.

2.5.7 Jeter un sort

Dans la classe `Game`, implémentez une méthode qui permet au joueur de jeter un sort qu'il a appris. Pour cela, on pourra afficher la liste des sorts appris (attention, cette liste peut être vide!). Par exemple, on pourra avoir les interactions suivantes :

```
Quel sort souhaitez-vous jeter ?
```

```
(1) Éclair en série
```

```
1
```

```
Vous avez lancé Éclair en série !
```

2.5.8 Quitter le jeu

Dans la classe `Game`, finissez l'implémentation de la méthode `start()` pour quitter le jeu si le joueur le souhaite.