Programmation avancée
TP 4 n° Donjons & Dragons – Modélisation avancée

V. Padovani, PPS - IRIF

Problème (janvier 2020)

On souhaite modéliser en Java les différents éléments d'un jeu de rôle simplifié (personnages, actions de ces personnages).

Chaque joueur choisit une classe de personnage parmi un éventail de possibilités (ici, quatre). Il choisit également une ou plusieurs sortes d'actions (ici, deux) parmi toutes celles que son personnage à la capacité d'effectuer, par exemple utiliser une certaine arme contre un joueur, ou encore lancer un certain sort contre un joueur.

Les effets des différentes sortes d'actions sont tous similaires : altération des points de vie de l'auteur de l'action (négative, positive ou nulle), et altération des points de vie de sa cible (idem). Seules les valeurs numériques de ces altérations diffèrent d'une sorte d'action à l'autre.

Le modèle importe ici plus que son implémentation. On ne vous en demande d'ailleurs ici qu'une implémentation très simplifiée : celle-ci devra permettre de créer les différents éléments du jeu, mais sans faire usage d'aucune structure de contrôle (ni `for`, ni `while`, pas même un `if`).

Règles du jeu

Choix du personnage. Il y a deux grandes catégories de personnages : guerrier, et initié. Un guerrier peut être soit un paladin, soit un voleur. Un initié peut être soit un sorcier, soit un moine. Chaque personnage reçoit en début de jeu d'un nombre certain de points de vie : 200 points pour un paladin, 150 pour un voleur, 100 pour chaque sorte d'initié.

Sortes d'actions. Outre le choix de son personnage, chaque joueur doit choisir pour celui-ci deux actions, une primaire et une secondaire, toutes deux choisies parmi deux grandes catégories d'actions : utiliser une arme, ou lancer un sort. Une arme peut être soit une épée, soit un gourdin. Un sort peut être soit un sort de lumière, soit un sort des ténèbres, soit un sort élémentaire. Les effets des actions associées à chacun de ces éléments sont décrits par le tableau suivant :

élément	*effet sur l'auteur*	*effet sur la cible*
épée	-10 pv	-30 pv
gourdin	+5 pv	-10 pv
sort des ténèbres	-5 pv	-20 pv
sort élémentaire	+5 pv	-10 pv
sort de lumière	aucun	+20 pv

Par exemple, utilisée sur une cible, une épée retire brutalement 30 points de vie à celle-ci, mais l'effort physique considérable déployé pour manier cette arme retire aussi 10 points de vie à son utilisateur. De même, un gourdin n'enlève que 10 points de vie à sa cible, mais l'effort physique modéré et salutaire déployé pour manier ce gourdin ajoute 5 points de vie à son utilisateur. Noter que le sort de lumière est neutre pour son lanceur, sauf bien sûr s'il est sa propre cible.

Choix des actions. Les choix possibles d'actions primaire et secondaire pour chaque sorte de personnage sont décrits par le tableau suivant :

personnage	*élément pour l'action primaire*	*élément pour l'action secondaire*
Paladin	épée	gourdin ou sort élémentaire
Voleur	épée ou gourdin	sort des ténèbres
Sorcier	sort des ténèbres	sort de lumière ou élémentaire
Moine	gourdin	sort de lumière

Modélisation du jeu

Exercice 1

Trouver une ou plusieurs hiérachies de classes permettant de modéliser le jeu, en gardant en tête les contraintes suivantes :

- Les personnages concrets seront crées en spécifiant via leur constructeur un choix d'actions primaire et secondaire, mais le modèle devra interdire *par typage* – *i.e.* dès la compilation – la création de personnages non conformes aux règles du jeu (*c.f.* ci-dessus, pas même de `if`).
- Le modèle devra permettre de simuler l'action primaire ou l'action secondaire d'un personnage, en prenant pour cible tout autre personnage (y compris lui-même), et en répercutant immédiatement sur les points de vie de chacun les effets associés à cette action.

Commencez par représenter graphiquement votre construction, en précisant dans le diagramme les classes abstraites, les champs et méthodes indispensables à chaque classe – pour les méthodes abstraites, précisez le niveau de leur implémentation.

Veillez à factoriser tout ce qui est factorisable. Noter la similarité des effets des actions : ce point est naturellement exploitable dans votre modélisation à l'aide de conversions ascendantes.

Exercice 2

Donner une implémentation complète du jeu[1]. Vérifiez dans un `main` la réussite de la cŕeation de personnages conformes au règles du jeu, et l'échec – en principe signalé par Eclipse dès l'écriture de ces tentatives – de la création de personnages non conformes.

1. Lénoncé original ne demandait qu'une implémentation partielle.