



LA ROCHELLE UNIVERSITÉ

LICENCE INFORMATIQUE

Génie Logiciel - Qualité

Logicielle et Gestion de Projet

TP3 - Cas d'étude "*Jeu de rôle*"

Application du design pattern Strategy

1 Comment réaliser ce TP

Pour réaliser ce TP, vous utiliserez l'outil Visual Paradigm disponible sur les VM Linux de l'Université (image ULR - Ubuntu 22.04). Vous pouvez également installer la version Community Edition sur vos machines personnelles. Le développement en Java sera réalisé sur l'IDE de votre choix.

Le dépôt de vos travaux dans un fichier `nom_prenom.zip` sera à réaliser pour **dimanche 6 avril 18h**. Les dépôts effectués par mail ne seront pas pris en compte et une note de 0 sera appliquée. Ce TP est à réaliser individuellement. Tout dépôt d'un projet qui ne compile pas sera noté 0.

```
nom_prenom.zip/  
├─ nom_prenom.pdf (contenant tous les diagrammes)  
└─ factory-vs-wild/ (contenant votre projet Maven)
```

2 Jeu de rôle - Description du cas d'étude

2.1 Rappel du cahier des charges

Dans le jeu de rôle *Factory vs. Wild*, les personnages et les monstres interagissent dans un monde dynamique. Chaque entité possède des caractéristiques spécifiques (vie, attaque, défense) et peut être affectée par des événements (prise de dégâts, état de santé, bonus d'équipement).

Les informations sur les personnages et les monstres sont enregistrées dans un fichier texte *characters_and_monsters.txt* dont voici un exemple :

```
# Format : Type; Nom; TypePersonnage; HP; ATK; DEF; Compétence
Personnage;Arthas;Guerrier;150;20;10;Coup de bouclier (Étourdit l'ennemi)
Personnage;Morgana;Mage;100;30;5;Boule de feu (Dégâts en zone)
Personnage;Legolas;Archer;120;25;8;Tir perçant (Ignore l'armure)
Personnage;Gandalf;Mage;130;35;4;Invocation d'esprit (Invoque une créature alliée)

# Format : Type; Nom; TypeMonstre; HP; ATK; DEF; Faiblesse; Résistance
Monstre;DragonRouge;Dragon;300;40;20;Glace;Feu
Monstre;OrcSanguinaire;Orc;180;25;10;Magie;Physique
Monstre;GolemDePierre;Golem;250;15;30;Eau;Physique
Monstre;SpectreNoir;Spectre;120;35;5;Lumière;Ténèbres
Monstre;Zombie;Mort_Vivant;100;10;5;Feu;Poison
```

2.2 (Re)mise en place du projet et vérification du travail

- Récupérez votre travail de la semaine dernière et vérifiez que votre projet Maven fonctionne toujours correctement.
- Récupérez le projet VisualParadigm de votre diagramme de classes. Le diagramme attendu à la fin du TP2 est donné en figure 1. Attention, les classes qui gèrent les exceptions ne sont pas représentées.

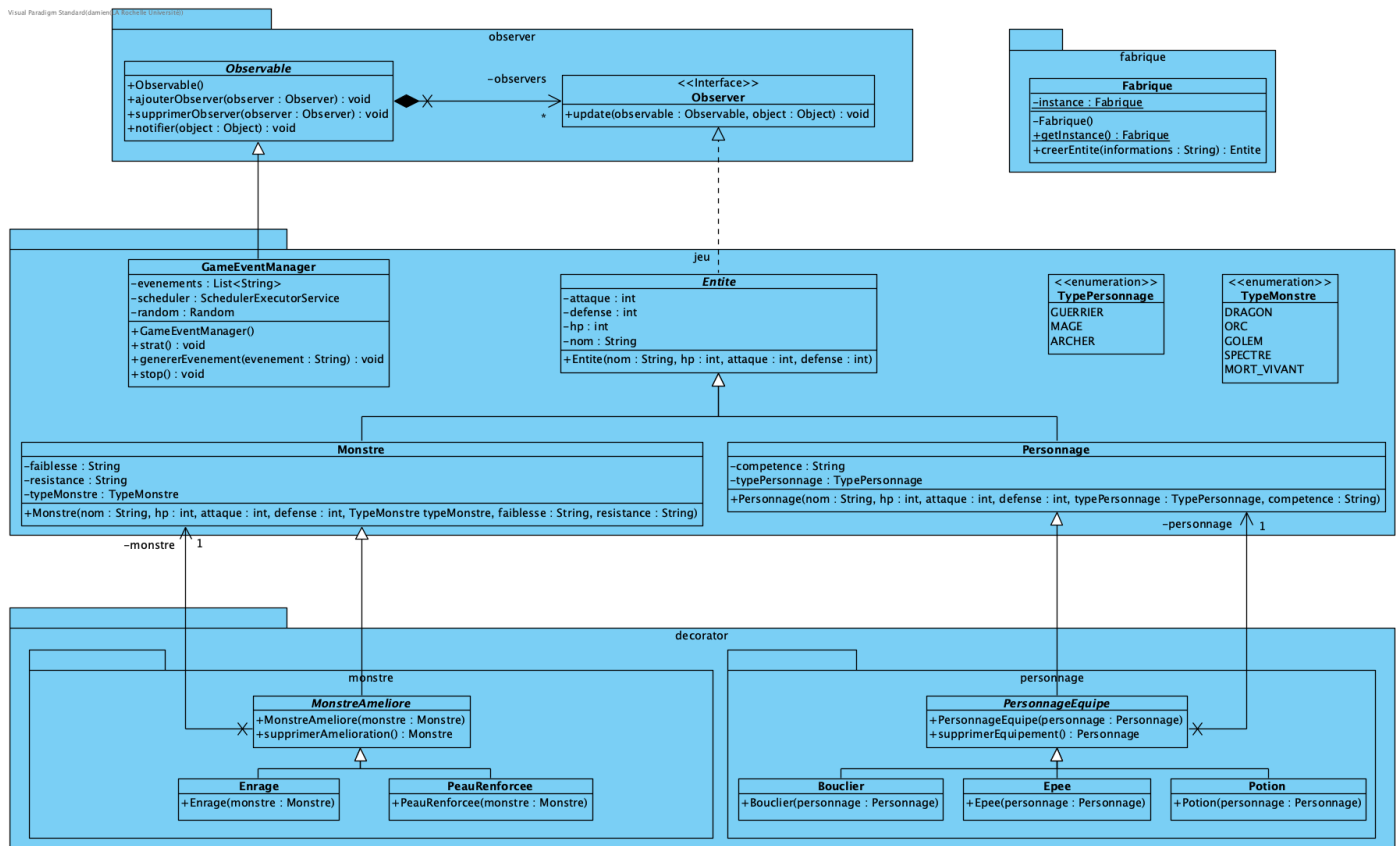


FIGURE 1 – Solution du diagramme de classes du jeu - v2

3 Mise en place du design pattern Strategy

Nous souhaitons pouvoir changer dynamiquement la stratégie d'attaque des monstres en fonction du contexte.

Soit $p \in P$ et $m \in M$ avec P l'ensemble des personnages et M l'ensemble des monstres. Les différentes stratégies possibles sont les suivantes :

Stratégie	Descriptif	Règle métier
Attaque à distance	L'attaque à distance consiste à frapper p sans entrer en contact direct avec lui.	p perd $\frac{8*m.attk}{3*p.def}$ points de vie et 1 point de défense.
Attaque de mêlée	L'attaque de mêlée implique un combat rapproché à main nue ou avec des armes.	p perd $\frac{15*m.attk}{2*p.def}$ points de vie et 2 points d'attaque.
Attaque normale	L'attaque normale est une attaque standard, équilibrée, qui ne dépend pas de la distance ni de conditions particulières. C'est l'attaque "par défaut" de m .	p perd $\frac{10}{p.def}$ points de vie.

TABLE 1 – Description des différentes stratégies

Chaque monstre peut utiliser différentes stratégies dans des conditions particulières décrites ci-dessous :

Monstre	Contexte		Stratégie
Dragon	Points de vie > 40 & distance > 15 m		Attaque à distance
	Points de vie ≤ 40		Attaque de mêlée
	Tous les autres cas		Attaque normale
Orc	Distance > 10 m		Attaque à distance
	Tous les autres cas		Attaque normale
Golem & Spectre	Tous les cas		Attaque normale
Mort vivant	Points de vie > 60	distance > 20 m	Attaque à distance
		distance ≤ 20 m	Attaque de mêlée
	Points de vie ≤ 60	Points de défense > 15	Attaque normale
		Points de défense ≤ 15	Attaque à distance

TABLE 2 – Description des différentes stratégies

- Modifiez votre diagramme de classes pour intégrer ce mécanisme au travers du design pattern Strategy. Vous ajouterez un nouveau package en conséquence.
- Implémentez ce design pattern en Java.

4 Défense des personnages

Les personnages n'utilisent pas de stratégie particulière. En revanche, ils peuvent se défendre. Les règles métiers de défense sont les suivantes :

Personnage	Stratégie du monstre	Règle métier
Guerrier	Attaque de mêlée	m perd $\frac{7*p.attk}{m.def}$ points de vie.
	Attaque normale	m perd $\frac{2*p.attk}{m.def*3}$ points de vie et $2 * p.attaque$ points d'attaque.
Mage	Attaque de mêlée	m perd $\frac{10*p.attk}{m.def*2}$ points de vie, 10 points d'attaque et 3 points de défense.
	Attaque normale	m perd 10 points de vie.
Archer	Attaque de mêlée	m perd 15 points de vie et 3 points d'attaque.
	Attaque normale	m perd $\frac{6*p.attk}{m.def*2}$ points de vie

TABLE 3 – Description de la défense des personnages

— Ajoutez cette mécanique de défense à votre code.