

ROYER Adrien
GASS Anthony



Rapport de Projet
Java POO
PvE RogueLike
(les fonctionnalités développées)

1. Introduction

Dans le cadre du cours de POO en Java, nous avons choisi de réaliser le deuxième projet proposé : “Jeu d’affrontement entre un joueur et des personnages”. À partir du sujet fourni, nous avons adapté et enrichi la logique de base pour créer un mini-rogue-like en mode console. Nous avons jugé nécessaire d’apporter plusieurs modifications et compléments au projet initial afin d’améliorer l’expérience de jeu et de mettre en pratique un ensemble plus riche de concepts (interfaces, héritage, collections, boutique, système d’équipement, évolution de classe, etc.).

Le joueur incarne d’abord un Paysan, réparti trois points de compétence (Force, Intelligence, Agilité), puis enchaîne jusqu’à dix combats contre des adversaires générés aléatoirement (Paysan, Barbare ou Magicien). Après chaque victoire, le joueur gagne un niveau et peut faire évoluer sa classe en Barbare ou Magicien dès qu’il atteint un certain seuil de compétences. Tous les deux combats, il accède à une boutique générant trois objets aléatoires : il peut en acheter un pour augmenter ses statistiques. Le jeu se termine lorsque le joueur meurt ou atteint le niveau 10.

2. Fonctionnalités développées

2.1. Création du personnage

Objectif: Permettre à l'utilisateur de donner un nom à son personnage, puis de répartir exactement 3 points de compétence entre les attributs Force, Intelligence et Agilité.

```
Quel est votre nom ?  
Faucon  
  
Vous devez placer 3 de compétences dans les catégories suivantes :  
Force / Intelligence / Agilité  
Points en force : 0  
Points en intelligence : 0  
Points en agilité : 0
```

2.2. Système de combat tour par tour

Objectif: Simuler un combat interactif tour par tour entre le joueur et un adversaire.

Fonctionnement :

- Affichage des PV et des statistiques des personnages à chaque tour.
- Choix d'action du joueur (attaque ou parade)
- L'adversaire choisit une action aléatoire parmi ses options.
- Les parades sont appliquées avant les attaques (dégâts réduits selon l'agilité).
- Les deux attaques sont résolues dans le même tour.
- Fin du combat si l'un des deux tombe à 0 PV.

```
Stats joueur :  
Nom: Faucon  
PV: 30  
Level: 0  
Force: 3  
Intelligence: 0  
Agilité: 0
```

```
Stats ennemi :  
Nom: Gorlag  
PV: 20  
Level: 0  
Force: 1  
Intelligence: 0  
Agilité: 2
```

```
--- Tour de Combat ---  
Faucon : 30/30 PV | Gorlag : 20/20 PV  
  
Choisissez une action :  
1. ATTAQUE_MELEE  
2. ATTAQUE_DISTANCE  
3. PARADE_MELEE  
4. PARADE_DISTANCE  
Votre choix : █
```

2.3. Système d'évolution du joueur

Objectif: Permettre au joueur de gagner en puissance, recevoir un point de compétence après chaque victoire, puis évoluer de Paysan vers Barbare ou Magicien quand il atteint un certain seuil de compétence (≥ 6).

Fonctionnement :

- À chaque victoire: +1 niveau, +5 PV max, +5 PV courants, +5 pièces d'or.
- Obtention d'un point de compétence à placer (Force, Intelligence ou Agilité).
- Dès qu'une statistique atteint 6, possibilité d'évoluer :
 - Paysan → Barbare (déploie sa force).
 - Paysan → Magicien (déploie son intelligence).
- L'inventaire et l'équipement restent actifs après évolution.

```
1
Vous pouvez évoluer en Barbare :
Tapez 1 pour évoluer
Tapez autre chose pour ne pas évoluer
1
Vous êtes désormais un barbare !
```

2.4. Boutique (Shop) tous les 2 combats

Objectif: offrir au joueur l'opportunité d'acheter des objets aléatoires pour augmenter ses statistiques tous les deux combats.

Fonctionnement :

- Génération de 3 objets aléatoires (type: casque, armure, bottes ou arme).
- Les statistiques des objets sont aléatoires et leur somme vaut 3.
- Le prix est aléatoire (entre 5 et 14 pièces d'or).
- Le joueur peut acheter un objet (s'il a assez d'argent), qui est automatiquement équipé.

```
1
-----Bienvenue dans le shop !-----
1 | botte 1 | Type : BOTTE | Force : 1 | Agilite : 0 | Intelligence : 2 | Prix : 12
2 | torse 2 | Type : TORSE | Force : 1 | Agilite : 0 | Intelligence : 2 | Prix : 5
3 | torse 3 | Type : TORSE | Force : 1 | Agilite : 2 | Intelligence : 0 | Prix : 7
Votre argent : 29
Entrez le numéro de l'item à acheter (ou '0' pour ignorer) :
█
```

2.5. Système d'inventaire et d'équipement

Objectif: gérer l'inventaire du joueur et appliquer automatiquement les bonus d'équipement à ses statistiques.

Fonctionnement :

- Inventaire stocké dans une HashMap (nom → Item).
- Un seul objet équipé par catégorie (casque, armure, bottes, arme).
- Les bonus d'objet sont ajoutés aux stats via des getters overrideés (Force, Intelligence, Agilité).

3.Conclusion

Ce projet nous a permis de mettre en pratique l'héritage, les interfaces et le polymorphisme en Java pour créer un système de combat tour par tour, un mécanisme d'évolution de personnage et une boutique d'objets. En combinant gestion d'inventaire, application dynamique des bonus d'équipements et génération d'un exécutable JAR, nous avons développé un mini-jeu cohérent, illustrant pleinement les concepts enseignés.