

# Rapport du Projet

## -Jeu Stratégique en Java -

**Filière: Ingénierie des Systèmes d'Information et Logiciels-L2**

**Année universitaire : 2025 / 2026**

**Module : Programmation Orientée Objet**

### **Groupe du projet:**

<u>Nom:</u>	<u>Prénom:</u>	<u>Matricule:</u>	<u>Groupe:</u>
Rahmoun	Hamida	232331336308	3
Sadi	Aymen	242431441503	3
Yazid	Aimen	242431441413	3

## **1. Introduction:**

Ce projet s'inscrit dans le cadre du module ‘Programmation orientée objet’ consiste à développer un jeu vidéo stratégique en Java. L’objectif est de mettre en pratique les concepts de la programmation orientée objet (POO) et de concevoir une architecture modulaire, extensible et conforme aux principes SOLID. Le jeu réalisé est un **jeu de stratégie au tour par tour**, où le joueur contrôle une faction, collecte des ressources, affronte des ennemis et progresse au fil des combats.

## **2. Contexte et Objectifs:**

**-Contexte :** application des notions de POO (héritage, polymorphisme, encapsulation) dans un projet concret.

**-Objectifs :**

- Concevoir un jeu jouable sur PC.
- Implémenter une architecture claire et extensible.
- Gérer une carte, des unités, des bâtiments et des combats.
- Respecter les contraintes techniques (Java 11+, collections, packages organisés).

## **3. Architecture du Projet:**

Le projet est organisé en trois packages principaux :

- **game** : contient la logique du jeu (boucle principale, carte, moteur).
- **entities** : regroupe les classes abstraites et concrètes des personnages (Player, Enemy, Character).
- **combat** : gère la résolution des affrontements entre le joueur et les ennemis.

### **Structure simplifié**

#### **game:**

-Main

-GameEngine

-GameMap

#### **entities:**

-Character

-Player

-Enemy

#### **combat:**

-CombatSystem

## **4. Description des Composants:**

### **4.1 Package game:**

- **Main** : point d'entrée du programme.
- **GameEngine** : boucle principale, gestion des commandes du joueur, exploration, repos, combats.
- **GameMap** : génération de la carte, placement du joueur et des ennemis, obstacles et affichage.

### **4.2 Package entities:**

- **Character** : classe abstraite définissant les attributs communs (santé, attaque, défense, position).
- **Player** : héritage de **Character**, ajoute l'expérience et la montée de niveau.
- **Enemy** : héritage de **Character**, ajoute le type d'ennemi, attaque aléatoire et esquive.

### **4.3 Package combat:**

- **CombatSystem** : gère les combats tour par tour, actions du joueur (attaque, défense, potion), coups critiques et résolution du combat.

## **5. Fonctionnalités Implémentées:**

- **Carte de jeu** : générée avec obstacles et ennemis.
- **Déplacement du joueur** : commandes Z/Q/S/D, avec vérification des obstacles.
- **Apparition d'ennemis** : aléatoire, avec types variés (Gobelin, Orc, Squelette, Bandit).

### **Système de combat :**

- Tour par tour.
- Actions du joueur : attaquer, défendre, utiliser potion.
- Ennemi avec attaque aléatoire et chance d'esquive.

- Coups critiques (25% de chance).

### **Progression du joueur :**

- Gain d'expérience après chaque combat.
- Montée de niveau avec augmentation des statistiques.

### **Événements aléatoires :**

- Exploration : possibilité de trouver un ennemi.
- Repos : récupération de PV, mais risque d'apparition d'un ennemi.
- Trésors : potions, armes, armures, or.

## **6. Résultats et Évaluation:**

Le projet est fonctionnel et jouable en console.

### **Points forts :**

- Architecture claire et bien séparée.
- Gameplay varié avec progression du joueur.
- Combats dynamiques grâce aux coups critiques et esquives.

### **Axes d'amélioration :**

- Gestion de plusieurs ennemis simultanés.
- Ajout d'un inventaire complet pour les objets.
- Interface graphique (JavaFX ou Swing) pour une meilleure immersion.

## **7. Conclusion:**

Ce projet a permis de mettre en pratique les concepts de la programmation orientée objet dans un cadre ludique et motivant. Le jeu réalisé constitue une base solide qui peut être enrichie par de nouvelles fonctionnalités (multijoueur, gestion avancée des ressources, interface graphique).

Il répond aux objectifs fixés et démontre une bonne maîtrise des notions de conception logicielle et de développement en Java.