

Cahier des Charges Synthétique : La Prophétie de la Mare

Projet : Conception et développement d'un jeu d'aventure (Console/Java)

Contexte académique : Projet UML - 3iL Ingénieurs 2025

Date de rendu : 21 décembre 2025

1. Présentation et Contexte

Ce projet s'inscrit dans le cadre du module de conception UML. Il vise à réaliser l'analyse, la conception et l'implémentation partielle d'un jeu d'aventure textuel.

Le jeu, intitulé "**La Prophétie de la Mare**", plonge le joueur dans un univers animalier fantasiste.

- **Le Lore** : Un village de canards est terrorisé chaque nuit par un renard qui se réfugie ensuite dans sa grotte.
- **Le Protagoniste** : Le joueur incarne le "Héros Canard", désigné par une ancienne prophétie pour mettre fin à cette menace.
- **L'Objectif** : Pénétrer dans la grotte (le donjon), survivre aux différentes salles et vaincre le renard logé au fond de celle-ci.

2. Utilisateurs Cibles

Le jeu s'adresse aux amateurs de **RPG textuels**, de *roguelikes* et de jeux rétro en ligne de commande. Il vise un public appréciant la réflexion (énigmes) et la stratégie au tour par tour (combats), sans nécessiter d'interface graphique complexe.

3. Spécifications Fonctionnelles

Le système doit permettre au joueur de naviguer dans un environnement hostile et d'interagir avec différents événements.

3.1. Navigation et Exploration

- **Système de Salles** : La grotte est modélisée par un ensemble de salles connectées.
- **Déplacement** : À chaque étape, le joueur peut choisir une direction (Haut, Bas, Gauche, Droite).
- **Contrainte de mouvement** : Pour simuler l'exploration progressive d'un donjon, le joueur ne peut jamais repartir immédiatement dans la direction d'où il vient (pas de retour arrière direct).

3.2. Typologie des Salles

Le système génère trois types d'événements prédéfinis lorsqu'on entre dans une salle :

1. Salle de Combat :

- Un ennemi est présent.
- Le combat est inévitable pour progresser.

2. Salle d'Énigme :

- Une question est posée au joueur.
- **Succès** : Le joueur gagne une récompense (soin, objet, passage).
- **Échec** : Le joueur subit des dégâts (piège).

3. Salle de Trésor :

- Contient un coffre que le joueur peut choisir d'ouvrir pour obtenir une récompense.

3.3. Système de Combat

Le combat se déroule au tour par tour via la console :

- Le joueur possède des Points de Vie (PV).
- À chaque tour, le joueur choisit entre :
 - **Attaquer** : Inflige des dégâts à l'ennemi.
 - **Se soigner** : Restaure une partie des PV.
- L'ennemi riposte automatiquement après l'action du joueur.
- **Condition de fin** : Le combat s'arrête si les PV du joueur ou de l'ennemi atteignent 0.

3.4. Conditions de Victoire et Défaite

- **Défaite** : Les PV du joueur tombent à 0 (lors d'un combat ou suite à une énigme ratée). Le jeu se termine (Game Over).
- **Victoire** : Le joueur atteint la salle finale et défait le Renard (Boss).

4. Spécifications Techniques et Contraintes

- **Langage** : Java (Standard Edition).
- **Interface** : Console de commande (CLI - Command Line Interface).
- **Architecture** : Le projet suivra une conception orientée objet stricte, favorisant le découplage entre la logique métier (le Système) et l'affichage (l'Interface).
- **Outils** : Modélisation UML, Git/GitHub pour le versioning.