

Rapport de projet : projet Java

1. Introduction

Dans le cadre du cours de développement informatique avancé orienté applications, nous avons été amenés à concevoir un projet sur le thème de notre choix. Nous avons choisi l'implémentation d'un jeu de plateau, inspiré en partie du jeu "Le Donjon de Naheulbeuk". Les joueurs seront confrontés à des ennemis dans un style PVE (*Player vs Environment*). Le but est de parvenir à éliminer des ennemis de salle en salle afin d'arriver au boss ultime et le vaincre pour gagner.

2. Rappel du cahier des charges

2.1 Les contraintes

- une architecture MVC, avec deux interfaces utilisateurs (l'une graphique, l'autre console) ;
- une communication Socket et/ou une gestion de base de données ;
- une structure de données du *framework* Java Collection ;
- une source GitHub pour le stockage du projet ainsi que le wiki ;
- Java comme langage de programmation.

2.2 Les attentes

Le jeu devra permettre à un joueur de :

- joindre une partie en coopération avec un autre joueur ;
- choisir un héros.

Notre application devra pour cela :

- gérer des rencontres joueurs contre ennemis ;
- créer des donjons (salles d'affrontements) ;
- créer des ennemis aléatoires dont le niveau croît en fonction de celui des joueurs ;
- gérer un système de *loot* (objet/arme que les ennemis morts laisseront aux joueurs) ;
- gérer un système d'inventaire pour les joueurs ;
- gérer un système de *leveling* (augmentation d'expérience après avoir terminé un donjon) ;
- créer une interface graphique.

3. Objectifs atteints

Au terme de notre projet, nous avons mis en place :

- deux interfaces, l'une console et l'autre graphique, selon l'architecture MVC ;
- une communication avec une base de donnée permettant de récupérer les *loots*, sauvegarder et charger une partie ;
- un choix entre quatre héros ;
- un multijoueur sur une même machine ;
- des rencontres joueurs contre ennemis ;
- un système de *leveling*, permettant aux héros ainsi qu'aux ennemis, de monter de niveau.

4. Fonctionnement du jeu

Lors du lancement du jeu, les joueurs sont invités à choisir un héros. Chaque héros, possède des caractéristiques différentes.

Lorsque les héros choisis, les joueurs entrent dans le premier donjon et sont confrontés à trois vagues d'ennemis. Chacun à leur tour les héros vont pouvoir choisir l'ennemi à attaquer. Une fois que les deux joueurs ont attaqués c'est au tour des ennemis. Lorsqu'un ennemi meurt les joueurs gagnent de l'expérience pour monter en niveau.

A la fin de chaque vague les ennemis morts laissent tomber des objets que les héros peuvent ramasser afin de gagner de l'expérience ou bien de changer leur arme si celle-ci est plus puissante.

Une fois arrivé à la dernière vague du donjon cinq, les joueurs vont devoir se confronter au *boss* du donjon. S'ils arrivent à vaincre le boss, le jeu est terminé et les joueurs ont gagné.

A chaque moment de la partie il est possible de quitter le jeu et grâce à un système de sauvegarde automatique, il est possible de reprendre sa partie précédente mais attention, si vous quittez la partie au milieu d'un donjon vous recommencerez au début de celui-ci.

5. Descriptif des classes

5.1 Personnage

Classe mère de *Héro* et *Ennemi*, elle reprend leurs caractéristiques ainsi que les méthodes communes à ceux-ci.

5.2 Hero

Classe fille de *Personnage*, celle-ci contient les caractéristiques et méthodes propres aux héros. Il y a quatre héros possibles, un *Elfe*, un *Nain*, un *Orque* et un *Humain*. Chacun de ces héros possède des caractéristiques propres, par exemple l'*Humain* a peu de vie mais fait le plus de dégâts contrairement à l'*Orque* qui fait peu de dégâts mais a beaucoup de vie.

5.3 Ennemi

Classe fille de *Personnage*, celle-ci contient les caractéristiques et méthode propres aux ennemis ainsi que les *boss*. Il y a quatre types d'ennemis différents, chacun possédant des caractéristiques propres à l'image des héros. Cette classe s'occupe également de créer les *boss* qui sont les ennemis ultimes des donjons. Encore une fois, chaque boss possède ses propres caractéristiques

5.4 Arme

Permet de créer des armes, qui sont celles utilisées par les héros et par les ennemis, ou ramassée sur des ennemis morts durant la phase de *loot*. Une arme a un nom et un nombre de dégâts.

5.5 Loot

Permet de créer des *loots*, ce sont les objets que les héros peuvent ramasser lorsqu'ils finissent un vague d'ennemis. Un loot est composé d'un nom et d'une valeur en expérience.

5.6 Donjon

Permet de créer les vagues d'ennemis que les héros vont affronter pendant le jeu. Les vagues d'ennemis sont générées de manière suivante, la vague une contient deux ennemis, la vague deux contient trois ennemis et la vague trois contient cinq ennemis. Les ennemis évoluent également en fonction du niveau des joueurs. Elle s'occupe de récupérer des *loots* de manière aléatoire dans une base de données.

5.7 Jeu

Le modèle de l'architecture MVC. Il supervise le déroulement du jeu et contient les méthodes permettant de sauvegarder et charger une partie.

5.8 JeuController

Le contrôleur de l'architecture MVC. Il reçoit les demandes de modifications des vues console / GUI, et modifie le modèle en fonction de celles-ci.

5.9 JeuVue

La vue de l'architecture MVC. Cette classe est parent de *JeuVueConsole* et *JeuVueGUI*. Contient le constructeur des vues et la connexion au modèle.

5.10 JeuVueConsole

Classe fille de *JeuVue*, celle-ci contient les méthodes relatives à l'affichage et l'interaction du jeu en interface console.

5.11 JeuVueGUI

Classe fille de *JeuVue*, celle-ci contient les méthodes relatives à l'affichage et l'interaction du jeu en interface graphique.

6. Conclusion

Nous avons pu répondre aux consignes du projet ainsi que nos attentes personnelles fixées dans le cahier des charges. Notre seul regret est de ne pas avoir pu afficher proprement les icones des joueurs et ennemis dans les mini *maps*, ainsi qu'un système d'inventaire avancé.

La plus grande difficulté rencontrée a été de réaliser un affichage propre pour l'interface graphique à l'aide de *java.swing*.

Ce projet nous a permis de nous familiariser avec le langage Java, l'orienté objet et comme nous avons choisi d'implémenter une base de données, de revoir la matière du cours de première, ainsi que de faire interagir du code source avec une base de données.

7. Ressources

<https://github.com/BJ2TL1/ProjetJava.git>

Table des matières

1. Introduction.....	1
2. Rappel du cahier des charges.....	1
2.1 Les contraintes	1
2.2 Les attentes	1
3. Objectifs atteints	2
4. Fonctionnement du jeu	2
5. Descriptif des classes.....	2
5.1 Personnage.....	2
5.2 Hero	2
5.3 Ennemi.....	3
5.4 Arme	3
5.5 Loot.....	3
5.6 Donjon	3
5.7 Jeu.....	3
5.8 JeuController	3
5.9 JeuVue	3
5.10 JeuVueConsole	3
5.11 JeuVueGUI	3
6. Conclusion	4
7. Ressources.....	4