BERNARD Geoffrey
LAGANE Alexandre
Binôme : B24

Projet de Programmation

JAVA

Université Paul Sabatié - 2014/2015

I. Introduction

Dans le cadre de nos études en L3 Informatique, nous avons eu l'occassion d'avoir à développer un projet de programmation (JAVA).

Le principe de l'application étant de confronter dans une arène les personnages de chaque équipe (étudiants de la licence) dans une arène. Nous devions donc réfléchir à des stratégies afin de remporter ce tournoi ! Une occassion d'avoir un apperçu sur le monde du travaille et de développer notre expérience dans le langage JAVA.

Ce rapport sera constitué de trois parties : une premiere partie explicant le sujet, une seconde partie détaillant la version locale et une dernière partie explicant la version collective.

II. Projet

1. Sujet

A partir d'un code existant, nous devions ajouter des Personnages et Potions ayant certaines caractéristiques et stratégies.
Nous avions à disposition un Personnage d'exemple, une arène, un serveur, une IHM, de quoi avoir un programme fonctionnel dès le départ.

L'accent était donc mit sur la réutilisation de code et l'application de nos connaissances JAVA acquise lors du premier semestre de notre 3ème année de Licence.
Le développement devait (de préférence) s'effectuer en binôme (Nous pouvions aussi choisir de le faire seul ou à trois).

Nous avions deux versions à faire : une version locale et une version collective.
Chaque version sera détaillée plus loin.

2. Choix

- Développement en équipe

Avec mon binôme, nous avons décider d'utiliser le gestionnaire de version nommé GIT afin de pouvoir développer notre application ensemble de manière simple et efficace. En effet, GIT permet de "partager" et de gérer les différentes versions du code.

- Stratégie du personnage de tournois

La stratégie utilisée pour le tournois est l'utilisation du charisme : 99 charisme et 1 de vie.
Notre personnage fuit les plus fort et tente d'aller en duel contre les personnages avec lesquels il gagne. Il tente des coups d'états afin de ne pas avoir de leader (Le gagnant étant celui ayant la plus grande équipe après 10min de jeu).

3. Personnages

Chaque personne a des caractéristique bien précises.
Dans la version **locale**, un personnage a (en plus d'un nom) une certaine quantité de :

- force : permet de tuer une autre personne
- charisme : permet d'ajouter un personnage à son équipe
- vie : 0 ou 1 qui dit si un personnage est en vie ou non
- leader : la référence de notre leader (-1 si on en a pas)

Dans la version **collective**, un personnage a (en plus d'un nom) une certaine quantité de :

- force
- charisme
- hp : nombre de points de vie du personnage
- vitesse : distance de vision d'un personnage (10 à 20)
- defence : permet de réduire les dégats subits (valeur de 0 à 60)
- leader

De plus, dans la version **collective**, afin qu'in personnage puisse arriver dans l'arène, il faut que ses caractéristiques réspectent l'équation suivante :

Force + Charisme + Vie + (10/60)*Defence <= 100

4. Potions

Chaque potion permet d'augmenter (ou abaisser) les caractéristiques d'un personnage qui la rammasse.

Exemple : une potion ayant 50 de force va rajouter 50 de force à la personne la rammassant.

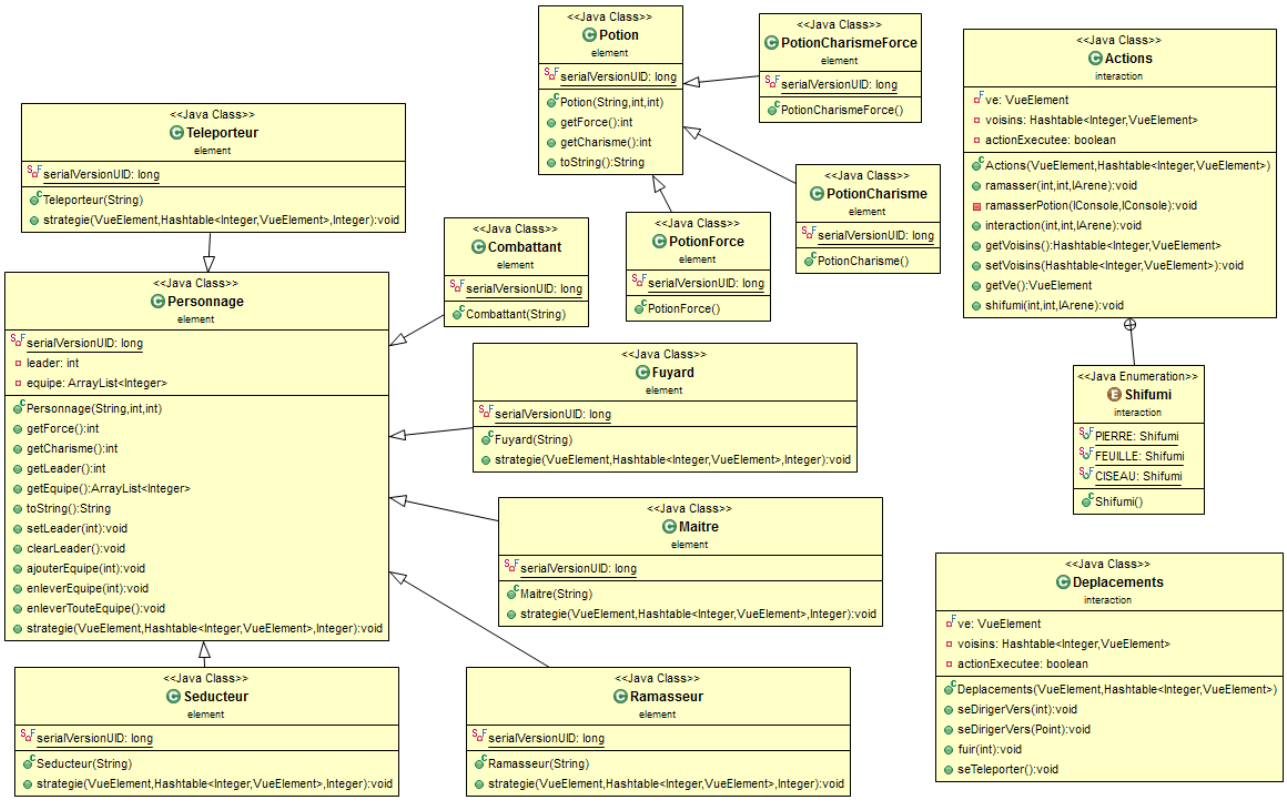
III. Version Locale

La version locale est une version dans laquelle nous avions le droit d'effectuer toutes les modifications de code souhaitées (dans toutes les classes du projet, nous n'étions pas limité).

L'objectif étant de créer au moins 5 personnages, 3 potions, d'ajouter quelques méthodes dans le package "interaction" (package gérant principalement les duels et les déplacements) et d'utiliser les méthodes afin de développer des stratégies de personnages.

1. Diagramme de classes

Nous avons utilisé le plugin *ObjetAid* afin de générer le diagramme de classes :



2. Personnages & Stratégies

- Combattant

Le combattant est un personnage qui effectue la stratégie de base (mise à disposition dans le projet fourni : il erre ou ramasse les potions ou effectue des duels s'il voit quelqu'un) mais possède plus de force qu'un personnage de base.

- Fuyard

Un fuyard a des caractéristiques relativement faibles. Il tente donc de fuir (sans sortir de l'arène) dés qu'il voit un ennemi autour de lui. Si il n'y a personne autour, il ne bouge pas.

- Maitre

Ce personnage a des caractéristiques très élevées et effectue la stratégies d'un personnage de base sans ramasser les potions (il n'en a pas besoin au vu de ses caractéristiques (max d'un *int signé*)).

- Seducteur

Le séducteur, s'il voit un ennemi avec un charisme inférieur au sien, va tenter de le prendre en duel, sinon il fuit la personne. S'il n'a pas de voisins, il erre.

- Teleporteur

Il reste immobile tant qu'il n'aperçoit pas d'ennemi. Lorsqu'il voit un ennemi, il se téléporte aléatoirement sur une case vide de l'arène. Lorsqu'il voit une potion, il la ramasse.

- Ramasseur

Le ramasseur erre à la recherche de potions et les ramasses dés qu'il est à portée. Il ignore (erre) les autres personnages.

3. Potions

- Charisme

Elle donne 50 de charisme lorsqu'on la ramasse.

- Charisme et force

Elle donne 50 de charisme et 50 de force lorsqu'on la ramasse.

- Force

Elle donne 50 de force lorsqu'on la ramasse.

4. Ajout de méthodes dans le package Interaction

Ajout d'une méthode **fuir** afin de fuir une reference donnée.

Ajout d'une méthode **seTeleporter** afin de se téléporter aléatoirement sur l'arène en allant sur une case vide.

Ajout d'une méthode **shifumi** afin d'avoir une autre facon de combattre.

5. Modification du package IHM

Nous avons remplacé l'affichage de base de l'IHM.

A l'origine les éléments de l'arène (Personnages & Potions) étaient représentés par des cercles (pleins) de couleur. Afin d'avoir une représentation plus jolie et de pouvoir distinguer facilement chaque element, nous avons appliqué une image à chaqu'un de ces éléments.

IV. Version Collective

La version collective est une version en réseau. Les modifications que l'on peut faire au code sont limitées.
Le but est de créer un personnage et de développer sa stratégie afin qu'il gagne la partie.

C'est cette version qui est utilisée pour le tournois. Chaque groupe a le droit de faire apparaitre un et un seul personnage dans l'arène de tournois.

- Personnages & Strategies

  - Fuyard

Le fuyard fuit continuellement vers l'éxterieur de l'arene (fuit un ennemi en priorité s'il en voit un).

Ce dernier gagnai assez simplement des points en fin de partie (il sort de l'arène et continu d'avancer vers l'extérieur, donc personne ou presque n'arrive à l'attraper) car la règle de base donnait des points aux survivant (après 10 minutes de jeu).

Or, la stratégie a été modifiée en cours de développement (l'équipe la plus grande remporte les points), c'est pourquoi nous avons ensuite développé un autre personnage : le Charismatique.

  - Charismatique

Le personnage Charismatique fuit les personnages avec lesquels il perd le duel.

Il essaye de convertir les personnages qu'il peut convertir (si le rapport de charisme le lui permet).

S'il se fait capturé (il a donc un leader), il essaye de faire des coups d'état (prendre en duel son leader afin de prendre sa place).

V. Conclusion

Ce projet nous a permis de mettre en pratique nos connaissances JAVA.
En effet, en plus d'avoir l'occassion de coder, nous avons dû lire, comprendre et utiliser un code déjà écrit par d'autres personnes.
Nous avons donc vu l'importance qu'ont les commentaires, sans lesquels il serait très long et complexe de pouvoir comprendre, maintenir et utiliser du code existant.

Ce projet était motivant car il aboutissait sur un tournois final entre étudiants. Cela nous a donc poussé à trouver une stratégie permettant de gagner la partie, et donc à développer un code performant.

Le fait d'avoir codé en binôme nous a permis de voir les avantages et "inconvéniants" de travailler en groupe.
Par exemple, trouver comment se répartir les tâches et comment réunir le code de chacun (solution retenue : utilisation du gestionnaire de version GIT).
De plus, si l'un de nous était bloqué, le fait de pouvoir s'appuyé sur son binôme permettait de trouver une solution plus rapidement et donc de développer plus rapidement.

Cette expérience a été bénéfique pour nous deux, aussi bien sur le plan personnel que professionnel.