

Projet Programmation 2

Première Partie

Louis Lemonnier
Stefan Schwoon

January 28, 2022

1 Introduction

Le Projet de Programmation 2 consiste cette année en l'implémentation en Scala d'un jeu du type "combat de monstres" en tour par tour, avec une interface graphique simple. La spécificité est que le joueur et l'adversaire peuvent avoir plusieurs monstres dans leur équipe, avec des caractéristiques différentes qui décident de la stratégie à adopter lors des matchs, ainsi qu'un certain nombre de capacités, qui peuvent baisser les points de vie de l'adversaire, baisser ses caractéristiques, ou augmenter celles du monstre allié, etc.

Le but du projet est d'utiliser les principes de la programmation orientée objet afin de gérer au mieux les différentes composantes du jeu. Vous travaillerez par groupe, en conservant à l'esprit que nos exigences seront adaptées à la taille du groupe.

2 Le jeu à programmer

La première partie du projet consiste à programmer un combat simple, l'interface graphique sera gérée par la librairie graphique de votre choix (par défaut, nous vous proposons d'utiliser Swing, toutefois il ne s'agit pas d'une contrainte).

2.1 Actions

Le combat doit consister en un tour par tour, avec choix d'une capacité à utiliser lors du début de chaque tour. Le monstre adverse lancera également une capacité lors du tour (ou non si raison spécifique). L'ordre dans lequel les deux combattants agissent se décide via une caractéristique que vous avez le choix de définir, tout comme le calcul des dégats, etc.

On ne demande pas une intelligence artificielle particulière dans cette première partie, les adversaires peuvent lancer leurs capacités aléatoirement.

2.2 Les alliés et les adversaires

Votre combat doit contenir de différents types de monstres alliés et adversaires. Lorsqu'un monstre perd tous ses points de vie, un de ses alliés le remplace, jusqu'à ce qu'il n'y en ait plus ; auquel cas votre jeu peut proposer un nouveau combat ou de s'arrêter.

2.3 Graphique

Pour la première partie il n'est pas demandé d'avoir un design graphique particulièrement esthétique, mais les interactions avec l'utilisateur doivent être intuitives. Les messages au joueur seront affichés par l'interface graphique et non pas dans la console.

3 Évaluation

3.1 Rapport et soutenance

Vous devez rendre un rapport de 2 à 3 pages et qui détaille vos choix techniques et les problèmes et difficultés que vous avez rencontrés. Dans le cas où certaines difficultés n'auraient pas pu être surmontées et que votre programme présenterait des défauts, vous pourrez expliquer ici ce que vous avez essayé d'entreprendre pour les résoudre et pourquoi cela n'a pas marché. Une soutenance de 15 minutes par groupe sera organisée à la fin de la première partie du projet où vous nous ferez une démonstration de votre programme.

3.2 Fonctionnalité du code

Votre projet sera évalué sur ses fonctionnalités. S'il remplit tout ce qui est demandé, rajouter d'autres fonctionnalités pourra apporter un bonus. La qualité graphique peut jouer un rôle, mais ce cours est avant tout un cours de programmation objet, ainsi, la perspective principale se portera sur les fonctionnalités, et l'évaluation de l'interface graphique reposera avant tout sur son caractère intuitif et facile à prendre en main.

3.3 Organisation du code

Votre projet devra être organisé de façon hiérarchique, et il vous faudra le séparer en fichiers, classes et méthodes. Il vous est recommandé de séparer le plus possible les différentes fonctionnalités, notamment les aspects *interface* (ang. *frontend*) et *fonctionnement du jeu* (ang. *backend*).

3.4 Qualité du code

L'évaluation prendra en compte la qualité de votre usage de la programmation orientée objet ainsi que la qualité de votre utilisation du langage Scala. Faites

attention à bien utiliser les propriétés d'héritages et à ne pas dupliquer du code inutilement. Utilisez plutôt des directives fonctionnelles que des boucles imbriquées. La mise en forme, la présence de commentaire et la cohérence des noms de classes, méthodes et variables devront être suffisamment décentes pour une lecture agréable du code.

4 Dates Importantes

- Deadline pour le rendu du code du projet : mardi 15 février 2022
- Date de la soutenance de la première partie : vendredi 18 février 2022