Projet Programmation 2
Troisième partie

Stefan Schwoon, Mathieu Hilaire

23 avril 2020

Le sujet de la troisième partie du Projet Programmation 2 est de développer des améliorations de votre programme faisant appel à des notions plus avancées de programmation. Comme tous les projets n'ont pas suivi les mêmes évolutions, vous êtes relativement libres dans votre choix. On vous propose la démarche suivante :

- Choisissez l'amélioration que vous souhaitez réaliser. Vous trouverez dans la suite quatre propositions, mais vous pouvez proposer autre chose si vous le souhaitez.
- Faites valider votre choix auprès des responsables du cours. L'entretien par visio-conférence sur la second partie est une opportunité d'en discuter.

Table des matières

1 Suggestions d'améliorations

De façon générale les améliorations proposées ci-dessus sont source de calculs plus intensifs, il devient vite nécessaire d'utiliser plusieurs thread en parallèles pour rendre le jeu fluide.

1.1 Affichage continu du mouvement des monstres et projectiles

Une première suggestion d'amélioration (pour ceux qui ne l'auraient pas déjà fait) est la gestion de l'affichage des monstres et projectiles au pixel près et non sur une grille. En actualisant l'interface graphique efficacement, vous pourrez donner l'impression d'un mouvement continu des monstres et projectiles même en présence d'un grand nombre de ces derniers.

1.2 Parseur de programme

Pour définir la séquence des monstres apparaissant à chaque vague voire les niveaux eux-mêmes, vous pouvez écrire un fichier texte et parser ce fichier

en Scala. Dans ce cas, vous utiliserez les combinateurs de parseurs de Scala qui permettent de gérer simplement des grammaires LL(*) arbitraires. La génération de monstres et les niveaux peuvent alors être définis avec des règles complexes.

1.3 Mode deux joueurs et réseau

Une amélioration consistante est d'implémenter un mode deux joueurs et d'utiliser le réseau pour faire communiquer deux ordinateurs différents. Vous pourrez utiliser les classes ServerSocket et Socket pour respectivement recevoir et envoyer les messages via TCP, comme décrit sur cette page :

http://stackoverflow.com/a/6416755/710358.

1.4 Stratégie collective des monstres et message passing

Notre dernière suggestion est l'implémentation de stratégies des monstres, où les entités ne disposent que d'informations locales mais peuvent communiquer via messages. En établissant un protocole à l'avance, il est alors possible de mener à une stratégie collective.

2 Critères d'évaluation

2.1 Rapport et soutenance

Les mêmes critères s'appliquent pour cette partie. Vous devrez rendre un rapport de 2 à 3 pages (en pdf généré par latex) qui détaille vos choix d'implémentations 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.

2.2 Fonctionnalités du code

Comme précédemment, votre projet sera evalué sur ses fonctionalités, dépendante de l'amélioration choisie.

2.3 Organisation du code

Les mêmes consignes qu'en partie 1 s'appliquent. Votre projet devra impérativement être organisé hiérarchiquement, en le séparant en fichiers, classes et méthodes. Vous tâcherez de séparer au mieux possible les différentes fonctionnalités en classes. Gardez sous le coude la règle de ne jamais avoir de fonction trop longue ou de fichier trop grand.

2.4 Qualité du code

De même, l'utilisation adéquate de la programmation objet et de Scala sera un critère important dans l'évaluation. Dupliquer du code dans des classes sous-entendrait une mauvaise compréhension de l'héritage. Préférez des directives fonctionnelles concises à des boucles for et if imbriquées, voir la Section 4.4 de l'introduction à Scala. 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.

3 Dates importantes

- Le code et le rapport seront à rendre avant le vendredi 29 mai à 23h59.
- Pour cette partie, au vu de la situation actuelle, on programmera des vision-conférences inviduelles selon disponibilités pour discuter des projets soumis.