

Projet Programmation 2

Troisième partie

Emile Contal & Stefan Schwoon

Le sujet de la dernière partie du Projet Programmation 2 est de développer des améliorations de votre programme faisant appel à des notions plus avancées de programmation. Comme vous ne suivez pas tous les mêmes cours de programmation, vous êtes relativement libres du choix de ces améliorations. Vous trouverez dans la suite quatre propositions d'améliorations. Si vous souhaitez implémenter autre chose, sollicitez nous pour que nous validions l'intérêt pédagogique. De façon générale ces améliorations sont source de calculs plus intensifs, il devient vite nécessaire d'utiliser plusieurs Thread en parallèles pour rendre le jeu fluide.

1 Suggestions d'améliorations

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 Écriture d'un parseur de programme

Pour définir la séquence des monstres apparaissant à chaque vagues voire les niveaux eux-mêmes, vous pouvez écrire fichier texte et parser ce fichier en Scala. Dans ce cas, vous utiliserez les combinateurs de parseurs¹ 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. <https://wiki.scala-lang.org/display/SW/Parser+Combinators+-+Getting+Started>

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 quelques pages expliquant les choix d'implémentation que vous aurez rencontrés. Dans le cas où votre programme présenterait des défauts, il faudra les mentionner ici en précisant leurs raisons. Une soutenance d'une dizaine de minutes par trinôme sera organisée à la fin de cette dernière partie du projet où vous nous ferez une démonstration de votre programme.

2.2 Fonctionnalités, efficacité et mise en forme

Comme précédemment, la lisibilité et la concision du code sera prise en compte dans la notation. Pour cette dernière partie, l'efficacité de votre programme sera un critère déterminant.

3 Dates importantes

- Le code et le rapport seront à rendre avant le mardi 17 mai à 23h59.
- La soutenance pour la première partie sera le vendredi 20 mai.