

Projet Programmation 2

Première Partie

YOAN GERAN

STEFAN SCHWOON

8 février 2024

1 Introduction

Cette année, le projet consiste à implémenter un jeu de gestion en Scala, avec une interface graphique simple.

Dans ce jeu, le joueur possède et gère une compagnie de train. Les trains se déplacent dans un monde avec des villes connectées par des chemins de fer, et transportent des passagers et des marchandises d'une ville à une autre.

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 de 2 ou de 3, en conservant à l'esprit que la notation sera adaptée à la taille du groupe.

2 Le jeu à programmer

Pour la première partie du projet, vous développerez une interface graphique basique et des fonctionnalités de base.

2.1 Graphique

La partie graphique du jeu n'a pas besoin d'être très développée. Le monde peut par exemple être représentée comme un graphe, avec des traits pour les routes et des carrés pour les villes. Pour l'instant, ce graphe, qui devrait consister de cinq à dix villes, sera identique lors de chaque instance du jeu, et les connections entre les villes ne changent pas.

Cependant on doit pouvoir visualiser certaines informations sur les villes (nom, population, passagers qui attendent, etc.) et les routes (longueur, etc.) par exemple lorsque ces derniers sont sélectionnés.

De même, la liste des trains possédés doit pouvoir être consultée (avec possiblement la possibilité d'avoir des informations sur chaque train), et la quantité d'argent doit être visible.

2.2 Actions

Il faut bien sûr pouvoir lancer une partie. Les autres actions attendues sont assez simples : il faut pouvoir créer un nouveau train (avec plusieurs types de trains possibles, par exemple pour le transport de marchandises ou de personnes, ou avec différents moteurs plus ou moins puissants), et créer un nouveau voyage entre deux villes.

Le jeu se déroule en temps réel. Ainsi, un voyage dure une certaine durée (en fonction du train et de la route). Lorsqu'un voyage est créé, des passagers peuvent embarquer, ce qui permet à la compagnie de gagner de l'argent. Le nombre de passagers dépendra de la ville (on peut par exemple avoir un nombre exact de personnes dans une ville, et faire ces nombres varier avec les différents voyages).

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 pourront 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 (frontend)* et *fonctionnement du jeu (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 12 mars 2024, à 20 h
- Date de la soutenance de la première partie : jeudi 14 mars 2024