

Projet Programmation 2

Première Partie

NICOLAS MARGULIES

STEFAN SCHWOON

30 janvier 2025

1 Introduction

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

Dans ce jeu, le joueur voit une carte où se trouvent des ressources, et doit placer des machines pour extraire lesdites ressources, les transporter et les utiliser pour fabriquer des produits plus complexes.

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. La carte peut par exemple être représentée comme une grille, avec des cases de couleurs différentes en fonction des ressources présentes sur la case, et des symboles (cercles, triangles, lettres) pour signifier la présence d'une machine sur la case. Pour l'instant cette carte sera identique lors de chaque instance du jeu.

Il devrait être possible de visualiser des informations supplémentaires sur une case (type de case, machines présentes et leur état, ect.).

2.2 Déroulé du jeu

Il faut bien sûr pouvoir lancer une partie. De plus, l'utilisateur doit pouvoir choisir parmi différentes machines disponibles et pouvoir les placer ou les enlever.

Il doit y avoir différents types de cases : des cases où on ne peut pas poser de machine, des cases vides et des cases où se trouvent des ressources. Une machine doit permettre d'extraire des

ressources de la case où elle se trouve, et une autre machine doit utiliser différentes ressources pour fabriquer quelque chose.

La plupart des machines ont une capacité de stockage de ressources limitée, et ne peuvent stocker qu'un seul type de ressources (ce qu'elles produisent). De plus, elles ne peuvent consommer que des ressources présentes sur des cases adjacentes (que ce soit au sol ou stockées par une autre machine). D'autres machines servent donc à transporter et stocker des ressources, et peuvent en partie s'affranchir des limitations précédentes (stockage de plusieurs types de ressources, capacité plus grande). Pour le transport, la machine de base est le convoyeur : il récupère des ressources des machines non-convoyeur adjacentes et les transfère si possible quelque temps plus tard à un autre convoyeur. Pour éviter les boucles, le convoyeur doit pointer dans la direction du convoyeur suivant. Différents types de convoyeurs pourront être utilisés pour mieux transporter les ressources (p.ex. un convoyeur qui répartit ses ressources entre deux sorties, ou des convoyeurs tunnel qui permettent d'occuper moins de cases, ou de faire se croiser deux lignes de convoyeurs).

Le jeu se déroule en temps réel, ainsi chaque machine met un certain temps à effectuer son action. Les machines fonctionnent en continu, et peuvent prendre un temps différent en fonction de la machine (voire en fonction de la ressource extraite pour la machine idoine). À chaque instant du jeu, chaque machine effectue son action par elle-même, et les machines communiquent entre elles au besoin en appelant leurs méthodes respectives, par exemple pour se transférer des ressources.

Le but du jeu (simplifié pour la première partie) est de produire le plus efficacement possible une ressource donnée (qui ne devrait servir à rien d'autre). Par exemple, on peut donner un temps limité pour la partie et compter la quantité produite à la fin.

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 : dimanche 2 mars 18h
- Date de la soutenance de la première partie : mardi 4 mars 8h