Réalisation d’un jeu en Java

ROBOT TURTLES

Théo Martinez – Timothée Pionnier



## Description du projet

Robot Turtles est un jeu de société ayant pour objectif de faire découvrir la logique de programmation aux plus jeunes. Le projet ici réalisé consiste à reproduire une version numérique de ce jeu en Java.

Description du jeu:

* Jeu de plateau pouvant être joué de 2 à 4 joueurs.
* Chaque joueur incarne une tortue qui se déplace sur le plateau avec pour objectif d’atteindre un joyau. Pour ce faire, le joueur créé une file d’instruction à partir des cartes de sa main, chacune permettant d’avancer, de tourner ou de détruire un obstacle.
* Un joueur peut donc à chaque tour choisir l’une de ces trois actions :
  + Ajouter une ou plusieurs cartes à sa file d’instruction.
  + Exécuter sa file d’instruction
  + Construire un mur afin de gêner ses adversaires
* Quelle que soit l’action choisie, le joueur peut décider de défausser toutes les cartes de sa main afin de piocher cinq nouvelles cartes.

## Architecture du projet

Le projet s’organise autour d’une classe principale : Partie.java. La classe Partie utilise l’implémentation BasicGameState de Slick2D.

C’est cette classe qui instancie le bon nombre de joueurs, eux-mêmes définis par une classe Joueur. Les cartes (déplacements, lasers, murs, main en cours et défausse) sont gérées par une classe Deck. A chaque création d’un objet Joueur, un objet Deck lui est attribué.

C’est également au sein de cette partie que la gestion des tours est effectuée. Dans ce GameState, le joueur en cours est capable de sélectionner l’action qu’il souhaite réaliser, ce qui l’amène dans le GameState correspondant. Une fois le tour terminé, le GameState Partie est rappelé, ce qui incrémentera le nombre de tours et passera la main au joueur suivant. S’il ne reste plus qu’un seul joueur en jeu, la classe Partie redirigera vers l’affichage des classements finaux (cf diagramme UML pour l’architecture complète du projet).

## Structure du projet

Le projet est structuré en 3 principaux packages :

1. **Package States : regroupe les états du jeu.**

* State Partie : État principal, il instancie les joueurs, redirige vers les states de Tours puis vers le state Winner si la partie est terminée.
* Package Tours : regroupe les 3 types de tours possibles.
  + State AddToProgram : Dans ce state, le joueur peut ajouter les cartes de sa main à sa file d’instruction.
  + State ExecProgram : Dans ce state, le joueur peut exécuter sa file d’instruction en cours.
  + State BuildWall : Dans ce state, le joueur peut construire un de ses murs sur le plateau.
* State Winner : La classe Winner récupère chaque joueur qui atteint le joyau au cours de la partie. Ce state affiche le classement final quand il ne reste plus qu’un joueur sur le plateau.

1. **Package Players : regroupe les classes utiles au joueur.**

* Classe Joueur : définit le type Joueur.
* Classe Deck : classe regroupant une pioche, une main, une file d’instructions, une pioche de murs et une défausse. Chaque joueur instancié de la classe Joueur possède son propre objet Deck.

1. **Package Objects : symbolise le matériel du jeu réel**

* Package Plateau :
  + Classe Plateau : définit le type Plateau. Un plateau génère 64 objets Cases.
  + Classe Case : définit le type Case. Une case possède une position sur le plateau et un état (symbolisé par un char).
* Classe Cartes : définit les relations entre un état de case (type char) et l’image affichée (type Image).