

Hanabi

Manuel du développeur



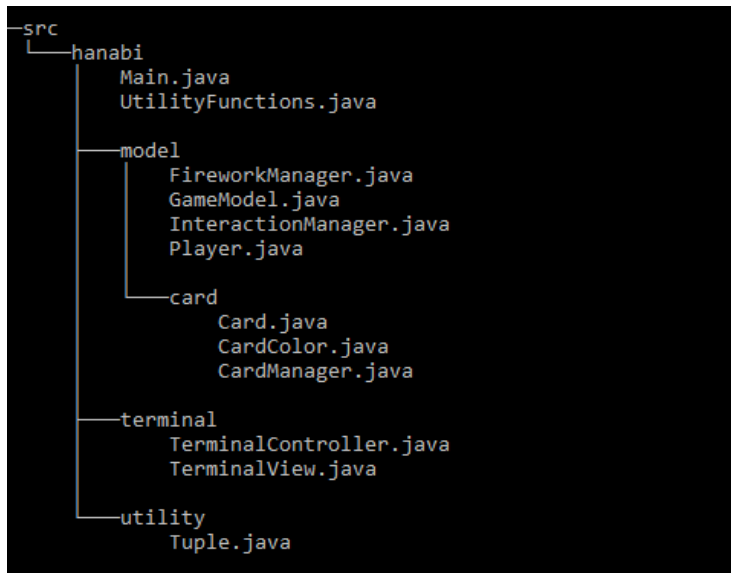
QUEMENER Lucas & EL HACHIMI Wassyl

Sommaire

Architecture.....	2
Fonctionnement	3
Évolutions	4

Architecture

Le jeu suit le modèle de développement MVC (Modèle-Vue-Contrôleur) :



Le package **model** permet d'initialiser les classes qui gèrent l'interactivité au sein du programme, avec `GameModel.java` qui est la classe la plus importante et contient toutes les fonctions de boucles et de gestion du jeu.

Le package **card** permet d'avoir un jeu de méthodes qui sont principalement utilisées dans `CardManager.java`. `CardColor.java` représente une énumération des couleurs de cartes.

On a un package **terminal** qui nous permet de contrôler l'affichage et permet de contrôler tout ce qui est en entrée, que ce soit des entiers ou des chaînes de caractères.

Le package **utility** contient une classe personnalisée qui nous sert à avoir quelques méthodes pratiques pour nos besoins.

Fonctionnement

Lors du choix du joueur, il doit taper littéralement "jouer", "jeter" ou "information" pour faire son choix, il est vérifié avec la méthode equals() si la chaîne est strictement identique ou non, tant que le joueur ne tape pas exactement une de ces chaînes, le programme continuera de demander l'action du joueur.

Ce principe est le même avec n'importe quelle action et étape demandée.

Le programme écrit littéralement chaque instruction que le joueur doit faire.

Évolutions

À l'heure actuelle, le jeu se fait intégralement avec le terminal, au prochain rendu nous aurons une interface graphique. On ajoutera également les extensions du jeu Hanabi afin de donner plus de possibilités aux joueurs.

Au niveau du fonctionnement interne du jeu, on pourrait faire l'intégralité du jeu avec le clic de la souris, par exemple au niveau du choix des cartes à montrer à un coéquipier, cliquer sur les cartes que l'on veut, détecter si l'action est légale, puis déterminer l'intention du joueur directement en fonction de ces choix, en l'occurrence, détecter quelle couleur il est en train de pointer ou quel chiffre il essaie de dévoiler.

On pourrait aussi ajouter des options qui permettrait de laisser la possibilité d'écrire une phrase quand on donne une information, au risque des joueurs.

Nous devons aussi optimiser l'architecture afin de faciliter la compréhension et l'immersion dans le code, et enfin modifier les visibilitées des classes et des méthodes à l'avenir afin d'assurer le bon fonctionnement du programme.