

Manuel Développeur

I. Définition du projet:

Hanabi est un jeu de société de type stratégie où plusieurs joueurs doivent collaborer pour réaliser les plus beaux feux d'artifice possibles. Le but de ce projet était de réutiliser les différentes structures et implémentations vues en cours, à travers la réalisation d'une version PC offline de ce jeu.

II. Mise en oeuvre:

Pour réaliser ce projet, j'ai conçu plusieurs classes en utilisant le pattern Modèle-vue-contrôleur en assurant la mise en place d'un interface graphique grâce à l'utilisation de la librairie zen5 dédiée à cet aspect de l'application.

Le package model lié à la logique du jeu contient les classes :

- AddCard
- Card
- ColorHint
- Discard
- Firework
- Game (la classe principale détenant les méthodes logiques de jeu comme donner une information, distribuer les cartes ...)
- GameAction (interface)
- GiveHint
- Hint (classe abstraite)
- NumberHint
- Player
- SimpleGameData (interface)
- l'interface Game.

L'interface SimpleGameData implémentée par Game ne contient que les éléments nécessaires à l'affichage de la Vue qui questionne l'état du jeu uniquement sans pouvoir le modifier. Cette fonctionnalité est déléguée au contrôleur.

Le package Command Line Interface C.L.I., contient les deux classes principales SimpleGameController et SimpleGameView, dont les méthodes de chacune d'elles modifient respectivement le Contrôleur et la Vue. La communication entre ces méthodes permettra un affichage du jeu en ligne de commande.

Le package GUI Graphical User Interface possède également ces deux mêmes classes dans l'idée d'afficher les composants du jeu sur une interface graphique. Il contient aussi le sous package components qui lui possède toutes les classes (dont une abstraite :

AbstractContainer) nécessaires à l'affichage des composants de la fenêtre de la Vue. Ce package components inclus ainsi, tous les composants d'affichage graphique en particulier la classe SplitPanel permettant d'organiser l'espace disponible de la fenêtre de la Vue.

Toutes les classes du package components implémentent l'interface UIComponent.

La classe SimpleGameView possédant un constructeur complexe (beaucoup d'arguments doivent être passés), nous utilisons le pattern de construction builder, pour dissocier la construction de Game avec sa représentation qui est la Vue. Ainsi le processus de construction du jeu aura la possibilité de créer via le contrôleur, des représentations de la Vue différentes.

Points d'attention :

Il aurait été judicieux d'utiliser le même mécanisme que celui du package GUI (à base de fonctions au lieu d'actions), ce qui permettrait de supprimer des classes dans le modèle, notées par le marqueur `@Deprecated`. En effet, celles-ci deviennent obsolètes (et ne devraient être utilisées) lorsque l'on fait des appels de fonctions qui en appellent d'autres.