

Projet Hanabi – compte rendu

Vous pouvez retrouver notre projet sur le site suivant : <https://github.com/Tsugaryu/Fireworks>

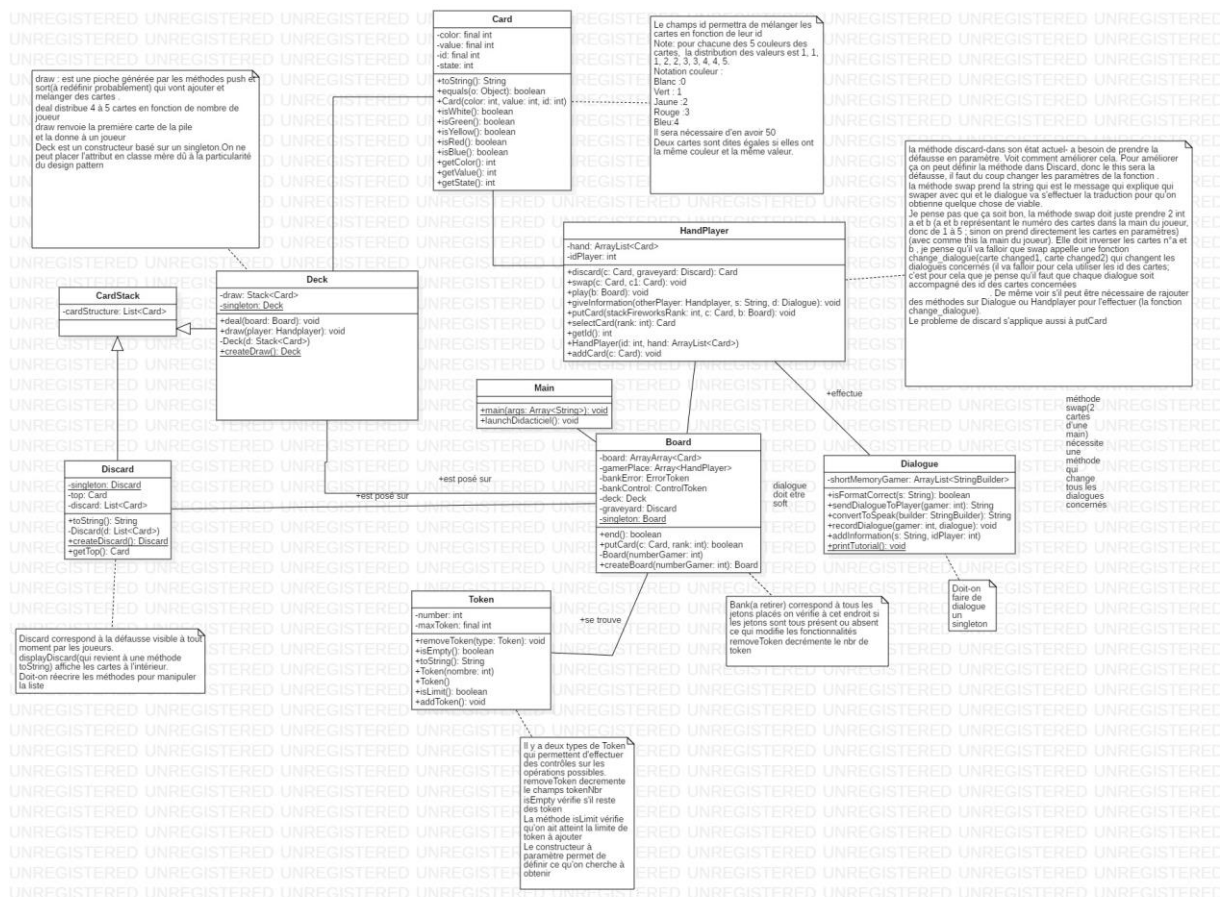
Notre projet Hanabi se compose en 10 classes réparties en 4 packages (se trouvant dans src/fr/upem/jeu/hanabi) :

- Le package game, contenant toutes les classes principales du jeu. Elles sont réparties en deux sous-packages :
 - Le package content, où se trouve les classes Card et Token, représentant respectivement le nombre de jetons à notre disposition, et les cartes du jeu. On utilisera deux instances pour la classe Token : une pour les jetons rouges, et une pour les jetons bleus).
 - Le package stack, où se trouve les classes Deck et Discard : ce sont des classes qui implémentent des listes chaînées de cartes qui sont utilisées comme des piles de cartes.

De plus, on peut retrouver dans le package game les classes HandPlayer - que l'on utilise pour représenter la main d'un joueur ; ainsi que Board, établissant l'état actuel du plateau de jeu. Board est la classe regroupant toutes celles du package game (et de ses sous-packages).

- Le package io possède les classes Dialogue, qui réalise l'interaction entre le système et les joueurs, et Parameter, classe que l'on utilise afin de choisir les paramètres du jeu → soit par défaut, soit en utilisant un fichier de paramètres déjà présent, soit en laissant l'utilisateur configurer son propre fichier de paramètres).
- Le package Main, où l'on peut retrouver la classe Menu (classe où se trouve la méthode menu permettant d'afficher le menu au lancement du programme, ainsi que certaines méthodes du jeu ; ainsi que la classe Main.

Voici le diagramme de classes que nous avons fait initialement :



Lors du lancement du programme, vous pouvez choisir 3 modes de jeu : le mode avec les paramètres par défaut, un mode avec des paramètres que nous avons choisis, et un mode où vous pouvez choisir vos paramètres.

Afin que notre code respecte la programmation objet, nous avons fait comme choix techniques de représenter chaque élément du jeu par un objet, que ce soit des éléments basiques comme les cartes ou les jetons, ou même les éléments plus avancés tels que la pioche (Deck) ou le plateau de jeu (Board), qui utilisent certains éléments de base.

De plus, il y a eu notamment 2 choix que nous avons dû faire :

- Dans Board, nous avons choisi de représenter l'état du plateau de jeu par un tableau de 5 Cards, et non par un tableau de tableau de Cards : en effet, avec le second choix nous aurions pu voir toutes les cartes qui ont été posées, cependant notre choix permet d'afficher seulement la dernière carte posée pour chaque couleur, ce qui est suffisant car si l'on a pu jouer une carte de valeur n, alors les cartes n-1, n-2, ... ont donc déjà été posées. Cela permet une économie de mémoire, mais aussi de faciliter les vérifications lorsqu'un joueur veut poser une carte.
- Dans Parameter, il fut difficile de choisir la manière d'écrire les paramètres dans le fichier : il fallait que le fichier soit lisible et facile à utiliser, mais aussi que la modification des paramètres ne soit pas trop compliquée à faire par l'utilisateur. Nous avons dû pour cela

utiliser un scanner ainsi qu'un buffered writer. Cela fut plus facile qu'utiliser des regex : nous aurions eu besoin de beaucoup de regex pour réussir à faire toutes les vérifications (et nous aurions pu facilement faire de nombreuses erreurs).

Lors du développement de ce projet, nous avons rencontré deux problèmes majeurs :

- Lors de la phase d'utilisation d'un fichier de paramètres, nous avons eu des problèmes lorsque nous voulions obtenir le fichier à cause de sa localisation.
- L'utilisation des regex (dans les classes Dialogue et Parameter) fut compliqué car il nous fallait vérifier correctement si ce que l'utilisateur entrait est correct ou non, ce qui a augmenté grandement la difficulté à cause de toutes les actions différentes possibles (que ce soit en jeu, ou pour la modification des paramètres).