

Rapport projet LO02

Le diagramme de classe finale n'est pas dans ce fichier word car il est trop grand et serait illisible. Il est en .jpeg juste au-dessus de ce fichier dans l'arborescence.

I/ Changements effectués entre la première version du diagramme et celle-ci :

- Ajout des vues Console et Graphique ;
- Ajout de la classe MDJController pour satisfaire le patron de conception MVC ;
- Ajout de la classe SaveData servant à la sauvegarde et au chargement de parties ; Ces sauvegardes et chargement sont aussi gérés par la classe ResourceManager ;
- Intégration du patron de conception Visitor proprement ; Au départ il était mal représenté ;
- Modification de la façon dont les trophées sont implémentés les trophées. Maintenant, plusieurs classes fille héritent d'une classe mère. On peut ainsi en ajouter si on le souhaite, ce qui rend le code plus modulaire.

II/ État actuel de l'application :

II-1 Ce qui fonctionne :

- L'application permet de jouer à plusieurs joueurs qui seront des joueurs physiques auxquels s'ajoutent des joueurs virtuels. On peut jouer à 3 ou 4 joueurs et faire n'importe quelle configuration joueur physique / joueur virtuel.

De plus, 3 stratégies sont disponibles pour les joueurs virtuels. Ces stratégies sont assignées aléatoirement. Ces stratégies sont : cacher sa meilleure carte et prendre la meilleure carte visible sur le plateau / cacher sa meilleure carte et en prendre une aléatoirement sur le plateau / cacher une carte aléatoirement et prendre la meilleure carte visible sur le plateau.

- L'ensemble de l'application est intégré dans une interface graphique. Les images utilisées sont celles données en annexe ainsi que d'autres que nous avons créés pour les variantes et l'extension du jeu.

- L'interface en ligne de commande fonctionne parfaitement. Elle est disponible dans la console. De plus, les actions faites dans la console sont liées à celles faites dans l'interface graphique (il faut d'abord configurer la partie : nombre de joueur, nom des joueur, mode de jeu, variant ou nom dans l'interface ou en ligne de commande).

- Nous avons utilisés pour l'extension les cartes 5, 6 et 7 auxquelles nous avons attribués aléatoirement un attribut trophée déjà existant. Nous avons créé ces cartes à partir de celles existantes. Avant le début d'une partie, il est demandé au joueur s'il souhaite intégrer l'extension ou non.

- En plus des règles de base, deux variantes sont intégrées. La première appelé « Reversed » consiste en un mode où la valeur des cartes lors du comptage des points est inversée (Par exemple le 4 de pic ne vaudra plus +4 mais -4 et inversement le 4 de cœur ne vaudra plus -4 mais +4). La deuxième variante implémentée appelé « Random » est celle où le trophée présent sur les cartes est distribué aléatoirement (par exemple le trophée du 2 de cœur ne sera plus avoir le joker mais celui d'une autre carte aléatoire). À chaque début de partie il est demandé au joueur quel mode de jeu il souhaite utiliser.

Le code est modulaire, il implémente correctement les trois patrons de conception que sont Strategy, Visitor et MVC. De plus, nous avons fait en sorte de pouvoir sauvegarder une partie puis de la recharger plus tard depuis l'interface graphique. Nous avons aussi fait en sorte que les actions faites dans la console au cours de la partie apparaissent dans l'interface graphique et inversement.

II-1 Ce qui ne fonctionne pas ou à améliorer :

Pour améliorer, il serait possible de rajouter des animations dans la vue graphique et de rendre le rendu dans la console plus lisible.