Architecture du projet

1) Sommaire

Dans ce rapport nous allons voir l'architecture choisi pour notre projet Splendor, projet visant à recréer le jeu de société Splendor dans une version se jouant dans la console de l'ordinateur. Nous allons d'abord détaillé le découpage du projet ainsi que les différentes classes, nous verrons ensuite sur quels points notre projet peut être améliorer mais nous verrons aussi que celui-ci à des points forts.

2) Découpage du projet :

Après lecture du projet nous avons décidé de le divisé en plusieurs classes que nous allons chacune détaillées.

La classe Cards : nous avons créé une classe dédiée aux cartes du jeu. Une carte étant défini par un nombre de point de prestige représenté par un entier, une carte possède aussi un prix pour cela nous avons implementer une Hashmap qui pour chaque couleur associe un entier correspondant au prix. Une carte possède aussi un entier correspondant à son niveau (il existe trois niveaux de carte).

Un enum color : nous avons créer un enum color listant toutes les couleurs disponibles de jetons dans le jeu.

La classe Noble : cette classe ressemble en tout point à la classe carte, nous avons seulement rajouter un champs string pour le nom de nos nobles. Dans cette classe noble nous avons implementer une fonction nommé isBuyable pour savoir à tout moment du jeu si un player peut recevoir la visite d'un noble.

La classe Player : Un player est défini par un nom, un age, une liste de jetons en sa possession, un nombre de point de prestige, une liste de cartes et une liste de nobles. Le player est initialisé grace à son nom et son age le reste des champs est initialisé à null. Dans cette classe player on retrouve les fonctions pour l'interaction au sein de la partie. On retrouve des fonctions pour connaitre le nombre de jeton du joueur en fonction de la couleur donnée en paramètre, des fonctions pour ajouter des cartes, des jetons ou encore des nobles au joueur. Nous avons aussi Override une fonction d'affichage pour que le joueur soit en connaissance de sa main à chaque instant et que la partie soit visuellement plus agréable.

La classe Board : La classe Board est la classe qui gère le plateau du jeu. Elle est composée d'une liste de joueur, un nombre de joueur, une liste de nobles, et trois Map: une Hashmap de cartes dans la pioche, de cartes sur le plateau et de jetons dans la banque. On retrouve aussi un boolean qui permet de gérer si le jeu va se déroulé en version simplifié (Phase 1) ou en version complète (Phase 2) Cette classe regroupe les fonctions par catégorie. La premiere catégorie contient les fonctions correspondant à la premiere action de jeu (Choisir trois jetons de couleurs différentes). La seconde partie elle est consacrée au fait de choisir deux jetons de meme couleur. Une autre partie s'occupe de l'affichage du jeu. Ensuite nous avons regroupé les fonctions utiles au jeu comme les fonctions d'intéraction avec le joueur, les fonctions d'ajout de cartes et de jeton. Enfin il y a deux grosses fonctions d'initialisation de la liste des cartes et des nobles fournis dans le sujet.

La classe Main : Pour terminer la classe Main initialise une partie et utilise la fonction game pour faire tourner le jeu puis retourner le vainqueur de la partie.

3) Points faibles du projet :

Nous avons conscience que notre projet comporte des faiblesses en vue d'un projet de programmation orientée object. Nous devrions surement créer une interfaces Card regroupant les cartes mais aussi les nobles car une carte et un noble se ressemble beaucoup. De plus notre projet n'ai surement pas découpé de la meilleure des façons. En effet quelques fonctions doivent pouvoir être déplacées dans des classes ou elles auront une place plus appropriée.

4) Points forts du projet :

Malgré ces quelques points faible notre projet peut aussi se venter de quelques points forts. Notamment sur le point de vu utilisateur, nous avons voulu rendre l'expérience agréable et le plus proche d'un vrai jeu de société. Les informations données aux joueurs sont donc les plus claires possible à chaque tour le joueur à une vision d'ensemble sur tous éléments. De plus nous avons controlé chaque saisie du joueur pour qu'une mauvaise saisie d'une couleur ou d'un nombre ne condamne pas sa partie ni celle des autres. De plus comme dans la réalité nous laissons la possibilité à chaque instant au joueur de revenir sur sa décision au cas ou il viendrait à changer d'avis. Aussi à l'exception de quelques fonctions la plupart de nos méthode toutes nos méthodes font moins de 20 lignes comme recommandé dans le sujet.

5) Conclusion :

Pour conclure nous pouvons dire que ce projet nous a permit de mettre en oeuvre ce que nous avons appris en cours et d'apprendre encore mieux à travailler en équipe.