Université de paris

Campus grand moulin

Rapport du projet : Langages à objets avancés

Réalisation d'un framework autour des jeux de cartes

Binôme :

Kawther **Djezzar**
Ny Andrianina Mamy **Razafintsialonina**

Table des matières

Introduction

Ce projet a pour but de réaliser une application permettant de jouer à plusieurs jeux de cartes. Cette réalisation consiste donc à concevoir un Framework, regroupant les similarités entre ces différents jeux, afin de faciliter l'implémentation de ces jeux et en appliquant les différents concepts étudiés tout au long du semestre avec le module : Langages à Objets Avancés.

I. Jeux choisis

En plus de la bataille dont l'implémetation a été exigée dans le sujet, nous avons choisis d'implémenter les jeux suivants :

1. Huit Americain

Le 8 américain est un jeu de cartes traditionnel dans lequel chaque carte a des effets sur le fonctionnement de la partie.

2. Uno

Le Uno est un jeu de cartes américain créé en 19711 par Merle Robbins et édité par Mattel. Basé sur les règles du 8 américain (qui se joue avec un jeu de cartes standard), le Uno lui applique un jeu de cartes spécifiquement dédié, ainsi que quelques règles supplémentaires.

3. Scopa

La scopa (en francais : le balai) est un jeu de cartes italien joué avec un jeu de cartes italiennes. C'est un des jeux de cartes les plus connus en Italie au même titre que la briscola. Il est joué également en Corse.

On y joue habituellement à deux joueurs, ou à deux équipes de deux joueurs, mais on peut aussi y jouer à trois, quatre ou six joueurs indépendants.

4. Briscola

Un jeu italien qui est un mélange entre scopa et bataille.

II. Architecture du framework

1. Classes implémentées

1.1 Classe Card

Comme son nom l'indique, elle a comme attribut la valeur et la famille de la carte, elle peut dans certain cas avoir un label comme dans le cas du joker.

1.2 Classe Deck

Représente un paquet, a comme attribut un tableau de cartes.

Permet d'éxecuter les differents opérations sur un paquet comme : Mélanger un paquet, Distribuer un paquet, Prendre une carte …

1.3 Classe DeckBuilder

La classe qui permet de configuer un paquet avec des cartes précises, tout dépend du jeu.

Cette classe utilise un design appelé Fluency qui est un chainage de fonctions (il sera détaillé en bas)

1.4 Classe Player

Représente un joueur, elle a comme attribut un paquet, un score et un nom.

On peut affecter des cartes à un joueur, et incrémenter son score.

1.5 Classe Board

C'est la classe la plus importante, celle-ci va gérer une partie de n'importe quel jeu, elle a comme attribut un tableau des joueurs, un paquet principal et un paquet temporaire.

Toutes les fonctions et les traitements nécessaires pour implémenter un jeu sont dans cette classe.

On cite par exemple : gérer le tour des joueurs, incrémenter le tour du jeu …

1.6 Classe GameTemplate

Tous les jeux héritent de cette classe, ou chaque jeu est implémenté par une classe qui a comme attribut un Board, et redéfinissant les différentes méthodes du GameTemplate qui sont :

Initialisation, first_turn, next_turn, is_the_end …

2. Éléments du C++ utilisés

1.1 Unique_ptr

std::unique_ptr est un pointeur intelligent qui possède et gère un autre objet via un pointeur et se débarrasse de cet objet lorsque unique_ptr est hors de portée.

On l'a utilisé pour éviter toute duplication de carte.

1.2 Fluency

Une désignation chaînée ou chaînage de méthodes (fluent pattern) consiste à agir en une seule instruction sur plusieurs méthodes du même objet, dans un but de plus grande lisibilité.

L'idée principale est qu'au lieu de qualifier chaque méthode par le nom de l'objet correspondant, ce qui conduisait à une lourdeur de style, ce nom d'objet reste par défaut actif dans toute l'instruction en cours.

Ce Fluent pattern a été utiliser dans DeckBuilder pour la construction d'un paquet en prenant en compte les numéros, les familles …

1.3 Iterator

Utilisé dans la classe deck pour itérer sur le vecteur des cartes, a fin de pouvoir réaliser les différents traitement, de prendre une carte au hazard, à une position précise, supprimer une carte, avoir l'ensemble des cartes…

1.4 Google Test

Google Test est une librairie qu'on a utilisée pour faire les tests unitaires.

1.5 Log Library

Utilisé beaucoup pour faciliter l'opération de débogage.

Conclusion

Au cours de ce projet dont le but est de concevoir une application permettant l'implémentation d'un ensemble de jeux de cartes, nous avons essayé de mettre en place les différentes connaissances acquises tout au long du semestre, cela nous a permis de renforcer ce que nous avons appris, que ça soit en dévelopement orienté objets, en programmation C++ et sur l'analyse et la conception.

Références

https://www.regles-de-jeux.com

https://www.regles.com/jeux-cartes/briscola.html

https://lucid.app/ (pour UML)

http://yunes.informatique.univ-paris-diderot.fr/accueil/enseignement/langages-a-objets-avances-cours-de-programmation-c