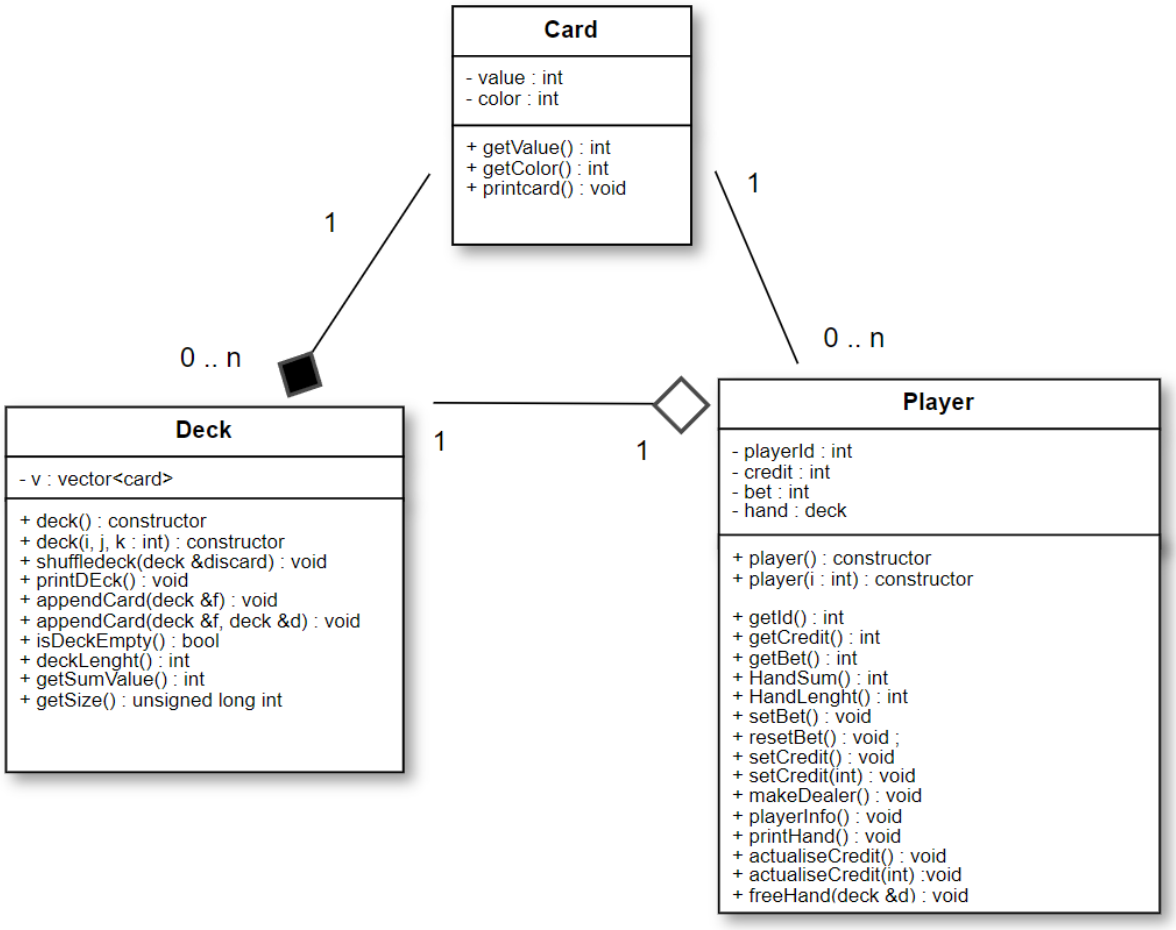

Organisation de la conception :

1. Compréhension du système de classe avec la réalisation de la classe Card. Lors de la même session, nous nous sommes penchés sur la question de l'affichage de deux choses : La valeur de la carte (Transformer la value 1 en 'As' lors de l'affichage permet d'éviter d'avoir une classe avec des chaines de caractères) et l'affichage du symbole avec l'Unicode. Même principe avec la valeur de la carte et du nombre de point qu'elle rapporte. Nous trouvons plus pertinent de faire une méthode qui converti la valeur en points plutôt que de surcharger la classe avec le nombre de point de la carte, un string pour son nom, un autre pour le symbole...

Nb. Concernant l'affichage de : {"♥", "♦", "♣", "♠"}, nous avons tenté dans un premier temps une fonction qui incrémenterai un code unique : u+2660 + color pour obtenir le bon symbole. Sans succès...

2. Réalisations des méthodes liées à la fonction carte : utilisation des constructeurs, des méthodes pour afficher les informations de l'objet. Enfin, début de la création d'un vecteur de carte et initialisation du deck de carte.

3. Réalisation des méthodes et fonctions liées au mélange du deck : nous initialisation les cartes dans la défausse, les mélangeons puis les basculons dans un nouveau deck : le playingdeck. Ceci est utile lorsque le playingdeck sera vide et que la défausse sera remplie : il suffira de rappeler la méthode de

mélange. Ainsi, nous gardons toujours 2 decks dédiées au jeu de cartes et la distribution de cartes sans en avoir un temporaire.

4. Création de la classe Deck qui est un vecteur de Card. Réalisation des méthodes liées à la manipulation des cartes au sein des decks, être capable de déplacer une carte d'un deck à l'autre, être capable de calculer le nombre de points présents dans un deck, le nombre de cartes dans le deck et ainsi de suite. Une fois ses méthodes réalisées, nous pouvons s'atteler à la création de la dernière classe : les joueurs.

5. Création de la classe Player : un joueur contenant un crédit, une mise et une main (Deck de Card). Cette classe contient les informations nécessaires à la bonne liaison des objets entre eux. Au sein de la classe Player, nous définissions une méthode pour faire d'un joueur le croupier qui sera tout le temps présent et sera techniquement le dernier « joueur » à rester à la table en fin de jeu. Ce sera le premier joueur de notre vecteur de Player, qui ne fait pas l'objet d'une classe, mais de fonctions liées aux mains.

6. Dernière ligne droite : création des fonctions liées jeu : initialiser les deck utilisés dans le main, initialiser un vecteur de joueur où nous renseignons les crédit, les mises, où nous commençons à distribuer des cartes. Une fois le début de la partie initialisé, c'est-à-dire :

- Les cartes sont correctement mélangées et disponibles.
- Les joueurs sont correctement initialisés et disponibles.
- Nous distribuons une carte au dealer, puis 2 cartes aux autres joueurs.

A partir de ce moment, nous pouvons commencer les tests concernant les choix des joueurs, les conditions et les branchements. Une fois que le joueur reçoit les bonnes options (par exemple un joueur n'a pas le choix de doubler la mise, si son crédit ne lui permet pas), nous pouvons s'intéresser au système de récompense de fin de partie. Enfin une fois tout cela réalisé, nous envisageons la mise en boucle, la fin du tour, l'expulsion des joueurs dont le crédit n'est plus suffisant pour participer au jeu…

7. Debug générale, tests, personnalisation de l'affichage. Tout au long de la conception et de l'implémentation, nous avons essayés de porter une attention particulière à certains détails tel que l'affichage de la mise de chaque joueur, de son crédit sans la mise, d'une bonne démarcation dans l'affichage et des actions des joueurs.