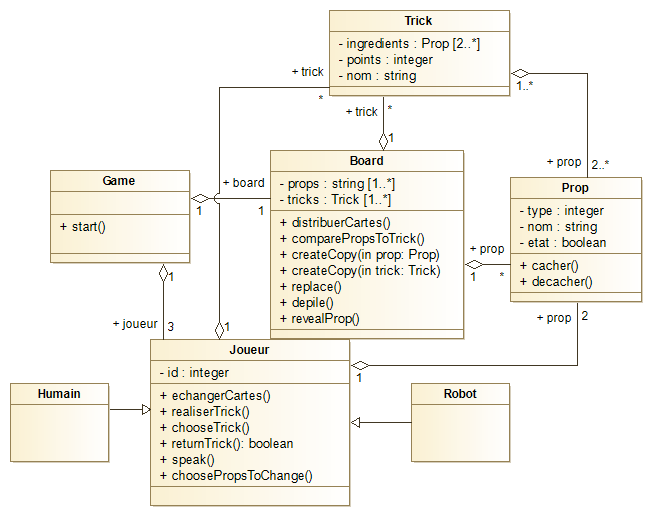
**LO02: Projet: The Other Hat Trick – Cantat\_Mallet**

**Compte Rendu final**

1. **Evolution du diagramme de classes**
   1. **Diagramme de classe originel**

* 1. **Diagramme de classe final et justification des évolutions**

**Une image contenant texte, carte

Description générée automatiquement**

Le changement principal, d’un point de vue modélisation, entre les jalons 2 et 3, qui a eu lieu est sûrement la suppression de la classe Board, censée initialement représenter le « plateau de jeux » et réguler les interactions des cartes entre elles, des joueurs avec les cartes, et des règles du jeux (la classe Game) avec ses instances. Pour donner suite aux conseils qui nous ont été faits lors du jalon 2, nous avons décidé de regrouper les deux classes Board et Game en une seule (Game), et de répartir les méthodes n’ayant attrait qu’aux cartes dans les classes des cartes : nous avons notamment repositionné dans la classe Trick la méthode qui compare les accessoires d’un joueur à ceux figurant sur une carte sort, initialement dans la classe Board.

Par ailleurs, nous avons également ajouté le patron de conception *Strategy* pour décrire et contrôler les différents comportements que le Robot peut adopter.

Nous avons implémenté un autre patron de conception, *Template Method*, pour les variantes jouables de notre jeu. Ainsi, nous avons créé 3 classes héritant de Game, chacune correspondant à une variante, et redéfinissant les comportements qui varient des règles de base.

Concernant les cartes, les instances des classes Prop et trick sont créées à partir des énumérations PropEnum et TrickEnum, qui nous permettent de recenser facilement le nombre de cartes dans le jeu. Cette solution de modélisation nous permet également de modifier le nombre de carte dans le jeu.

1. **Etat actuel de l’application**

Nous avons implémenté le patron de conception MVC, avec une vue graphique et une vue texte. Les deux vues implémentent l’interface Observer. Contrairement à ce qui était demandé, nous ne pouvons pas jouer avec les deux vues de manières concurrentes. Nous avons implémenté une vue graphique jouable, avec un Menu pour initialiser une partie. Le contrôleur récupère les informations fournies par la vue graphique et met à jour le modèle. Le modèle notifie les vues. La vue texte affiche simplement l’ensemble des actions réalisées par les joueurs mais ne peut récupérer aucune information.

Les variantes sont jouables avec la vue graphique, et on a ajouté un délai entre chaque étape pour pouvoir suivre les actions des robot (sans lesquels les mise à jour de la fenêtre seraient trop rapides pour être suivies).