

## UML MU4RBI01 Projet "1000 bornes"

### Concepts abstraits : Classes abstraites + héritage

Joueur, JoueurHumain hérite de Joueur, JoueurAI hérite de Joueur

### Concepts concrets : Classes concrètes

MilleBornes détient et partage des objets avec Cartes et Joueur

Cartes est une agrégation de MilleBornes

Joueur est une agrégation de MilleBornes

Joueur contient une référence à plusieurs objets de Cartes

### Méthodes abstraites

JouerCarte, TypeJoueur sont des méthodes abstraites dans la classe Joueur

La classe abstraite Joueur permet de définir l'architecture des classes filles pour le joueur humain et IA. Nous avons défini ces deux classes (JoueurHumain et JoueurAI) pour que l'on puisse décrire la même méthode mais qui définissent deux cas différents. La première étant s'il faut gérer en entrée un utilisateur ou bien implémenter une méthode qui fera un choix par l'ordinateur.

La classe Carte est chargée de créer les cartes du jeu (instances de classe de Carte) du jeu à l'initialisation. La classe Carte a un nom et un type (attaque/défense/botte). On ne souhaite pas que l'attribut de type PaquetCarte dans Cartes soit modifiable depuis l'extérieur. On utilise un attribut et une méthode privée.

La classe milleBornes détient un objet de type carte ObjCarte et une liste de joueurs. Cette classe permet de gérer les tours de jeu et le joueur demande une carte au jeu. La carte est ajoutée dans la main du joueur par l'attribut de cartesEnMainJoueur. Le joueur joue une carte de sa main sur lui-même ou sur un adversaire. Le score évolue à la fin de chaque tour de jeu

La classe JoueurAI permet un choix de carte qui sera fait par l'ordinateur. Nous avons implémenté plusieurs méthodes pour nous permettre de suivre les règles du jeu tout en automatisant le choix des cartes au fur et à mesure de la partie. La méthode updateScore nous permet de palier au problème que le joueur doit faire exactement 1000 bornes pour remporter la partie. Arriver à un score supérieur à 800, la carte distance ne peut plus être posé et ainsi de suite jusqu'à arriver au score final. La méthode AI\_niveau1 permet de nous défendre, de poser des cartes distances et également de défausser quand cela est nécessaire.

La classe JoueurHumain possède une méthode help qui permet de faire des suggestions au joueur avec les règles du jeu. Prenons un exemple, si un adversaire a posé une carte attaque, elle va nous proposer quelle carte on peut poser dans le cas où on a choisi une mauvaise carte.