



1^{re} YEAR OF MASTER
PYTHON PROJECT REPORT

Jeu de 1000 bornes

Groupe :
Morteza KAZEM DEHDASHTI

Enseignant :
Thomas Dietenbeck

Dec 2020

1 Les Relations entre les classes

Le jeu de 1000 Born possède neuf classe différents que je vous explique dans la suite :

1.1 Class Principal

: Le class principal est une classe pour préparer l'initial de jeu comme le nombre des joueur et leurs noms, Ensuite il définit un attribut de la classe **Jeu** pour initialiser le jeu, le démarrer et à la fin de jeu il récupère le résultat et l'affiche la personne gagnent. Donc la seule relation qu'il a est avec la classe **Jeu** .

1.2 class Jeu

Class **Jeu** contient plusieurs joueurs donc la relation entre class **Jeu** et **Joueur** est **Composition** comme la relation entre **Hôtel** et **chambre** , le **Jeu** à composer aux **Joueurs**. Cela a défini dans l'attribut de joueurs qui est un tableau de type **Joueur**.

La relation entre la class **Jeu** et **Piocher** est association parce que la classe **Jeu** utilise à la class **piocher** pour prendre 6 **cart** random et les donne aux joueurs. La même relation entre classe **Jeu** et la classe **Cart** , La classe **Jeu** utilise à la classe **Cart** dans le cas où il veut valider le **cart** ou récupérer le type de **cart**.

1.3 class Joueur

La classe **Joueur** utilise à la classe **Piocher** pour récupérer une **carte** random donc la relation est association. Entre class **Joueur** et la classe **Cart** il y a deux type de relation :

Aggregation : Chaque **Joueur** à une série de **cartes** à la main qui est à sauvegarder dans le tableau de **mycards**

Association : J'ai défini une relation association parce que class **Joueur** utilise à la classe **Carte** pour récupérer le type de **carte**.

1.4 class Cart

La classe **Cart** est une classe de type abstrait avec les méthodes abstraites. Les classes **Born**, **Botte**, **Parade** et **Attaque** ils ont tous héritent à cette classe et ils ont implémenté les méthodes abstraites de leur mère. J'ai choisi de définir class **Cart** comme la base de ces autres classes parce qu'ils ont tous des caractéristiques communes (Les méthodes des fois avec le même corps des fois avec les différents corps) donc j'ai défini une classe abstraite que les autres héritent cette classe.

1.5 Execution(Windows)

Phyton **Jeu.py**