



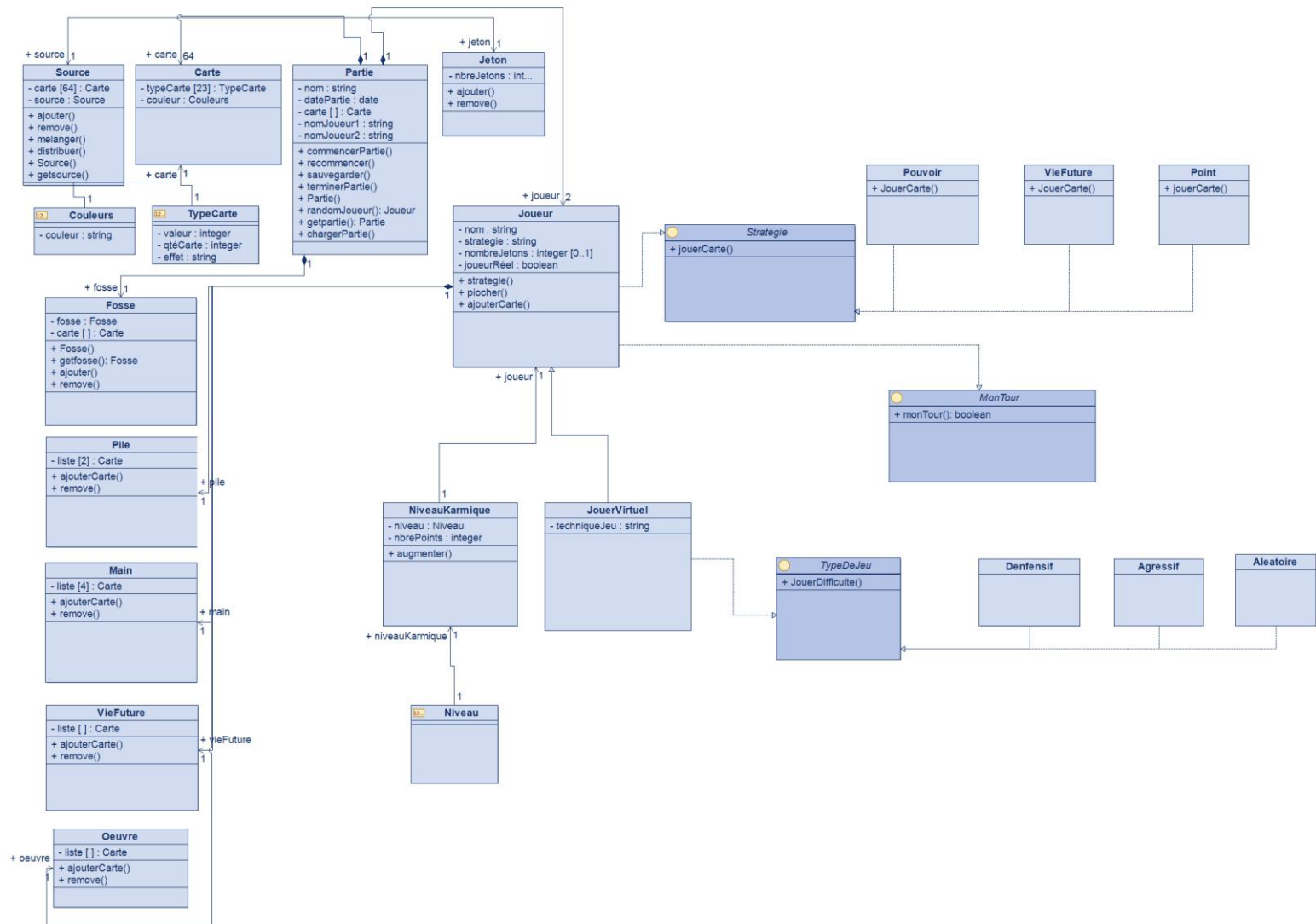
LO02 : PRINCIPE ET PRATIQUE DE LA PROGRAMMATION ORIENTE OBJETS

Livrable 2 KARMAKA



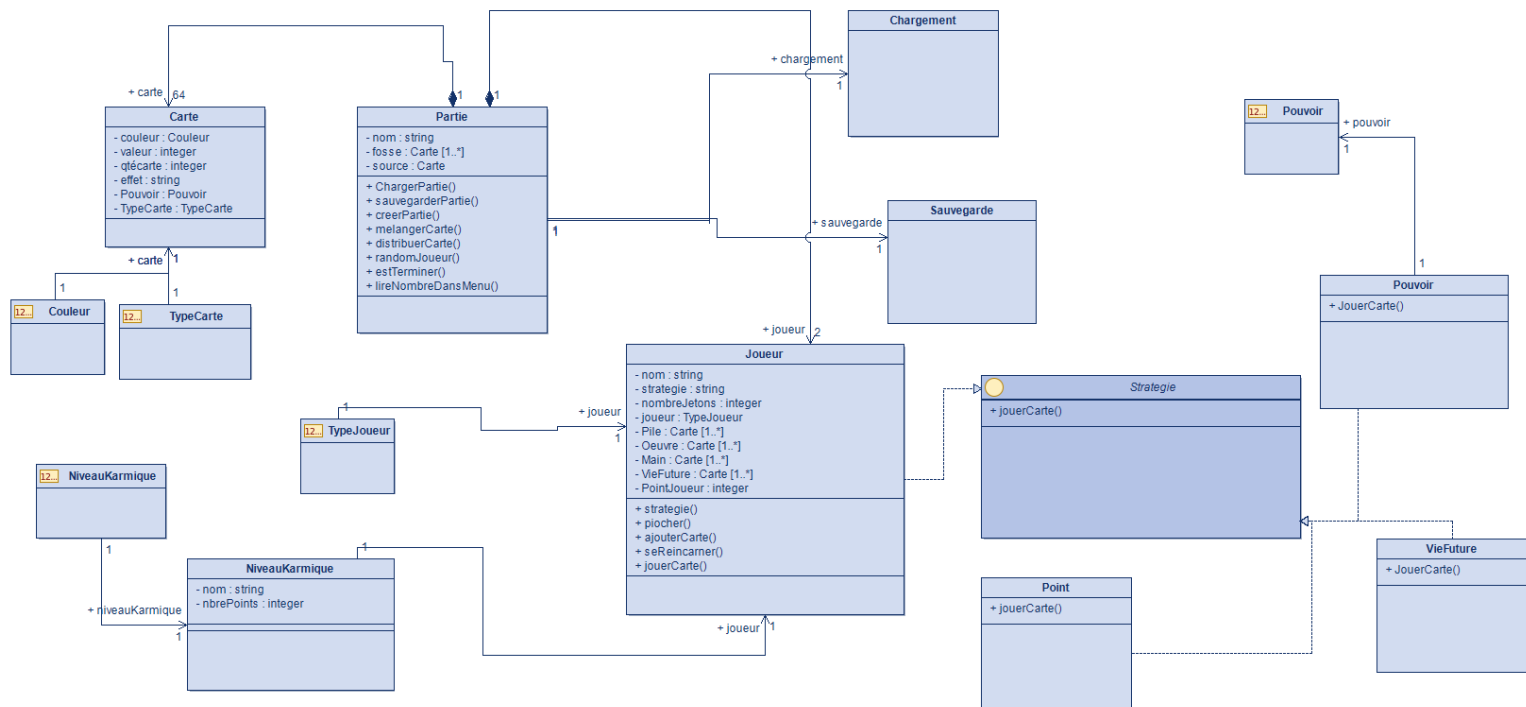
Patrice Mbangue, Vigny Brayan TAKAM TALLA

I- DIAGRAMME DE CLASSE AVANT LA PRODUCTION DE CODE



II- DIAGRAMME DE CLASSE APRES REALISATION DU PROJET ET MODIFICATION APPORTEE

A- Diagramme de Classe



B- Modification Apportée

- Nous avons supprimé l'interface **MonTour** prévue pour la gestion du tour des joueurs pour utiliser des attributs permettant de le gérer facilement.
- Nous avons supprimé la classe **joueur virtuel** afin de pouvoir la gérer dans la classe **joueur**, grâce à un attribut qui va caractériser le type de joueur (c'est une énumération comportant le type **Joueur Réel** et **Joueur Virtuel**)

- La stratégie du Joueur virtuel à été supprimé afin que les deux Joueurs implémentent la même stratégie
- En ce qui concerne les classes : Fosse, Main, VieFuture, Œuvre, nous les avons supprimés afin de créer une liste(collections) de chacune de ces classes dans un Joueur pour une manipulation plus rapide et plus facile
- En ce qui concerne les classes : Source et Fosse, nous les avons supprimés afin de créer une liste(collection) de chacune de ces classes dans une Partie pour une manipulation plus rapidement et plus facilement
- Nous avons supprimé la classe jeton afin d'utiliser un attribut caractérisant le nombre de jetons d'un Joueur à un instant donné
- Nous avons rencontré un problème lors de la sauvegarde de la partie car elle contenait des attributs statiques qui empêchaient sa sauvegarde complète. Pour pallier à ce problème, nous avons créé une classe Partiesave qui va stocker l'ensemble des attributs statiques et non statiques.

C- Fonctionnalité

Parlants des fonctionnalisées du projet, nous avons effectué toutes les spécifications liées au fonctionnement du jeu Karmaka, malheureusement nous n'avons pas pu finir d'implémenter l'interface graphique car il fallait modifier considérablement le code pour se pencher au modèle MVC.