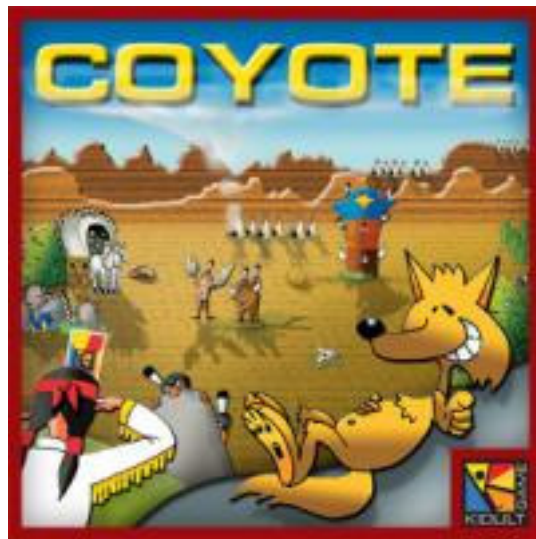


Le jeu du coyote

Etude conceptuelle UML



Sommaire

Introduction

I. L'architecture client-serveur.....	4
II. L'étude conceptuelle UML.....	5
III. Insertion des liens d'héritage.....	12
IV. Diagramme des classes généralisé UML.....	13
V. Persistance des données.....	14

Conclusion

Introduction

Le jeu du coyote est un jeu de société au départ qui va être analysé pour devenir un jeu en réseau. Les règles du jeu du coyote sont simples.

Les cartes sont tout d'abord mélangées, puis chaque joueur en reçoit une qu'il ne regarde pas. Il la coince entre le bandeau et sa tête, de façon à ce que tous les autres joueurs puissent facilement voir le numéro inscrit dessus.

Le premier joueur commence par annoncer un nombre de son choix (obligatoirement positif), puis son voisin a alors 2 possibilités :

1. Il annonce un nombre plus grand
2. Il pense que le joueur a annoncé un nombre supérieur à la somme de toutes les cartes que les joueurs ont sur le front (y compris lui-même), et crie "Coyote" !



Tous les joueurs déposent alors sur la table les cartes qu'ils avaient sur le front. Puis, on calcule le score total des cartes, en tenant compte des effets des cartes spéciales, telles que la carte "-Max" qui annule la carte avec le plus grand nombre d'une carte positive, la carte x2, qui multiplie par 2 le score, etc... Tous les effets sont cumulatifs.

Si le joueur avait effectivement annoncé un nombre trop élevé, il prend un pion coyote. Si ce n'est pas le cas, c'est celui qui avait crié "Coyote" qui prend le pion.

Lorsqu'un joueur a 3 coyotes sur son front, il est éliminé. Les autres continuent jusqu'à ce qu'il n'en reste qu'un.



I L'architecture client-serveur

L'architecture utilisée est une architecture client-serveur centralisée. C'est-à-dire que l'on a un serveur central et des clients (utilisateurs) qui se connecteront à celui-ci. Le serveur central contiendra tous les objets de chaque partie ainsi que la base de données nécessaire pour la persistance des données.

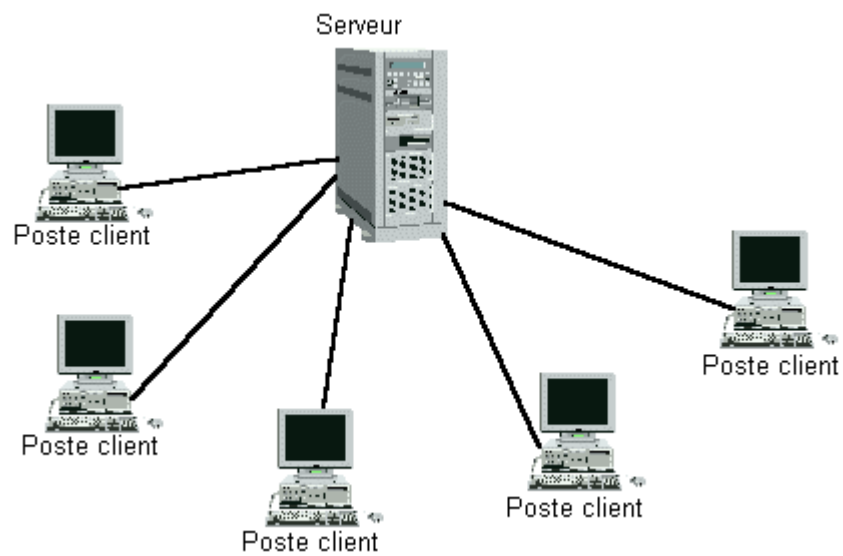


Figure 1 : Architecture client-serveur

II L'étude conceptuelle UML

1. L'action "Générer une pioche "

Les premiers points qui sont abordés concernent l'action "générer un jeu de cartes". Cette action intervient au début de chaque partie. Pour cette action, le diagramme des cas d'utilisation a été réalisé. Le lien de collaboration permet de connaître les objets qui seront en mémoire lors de l'exécution de l'action.

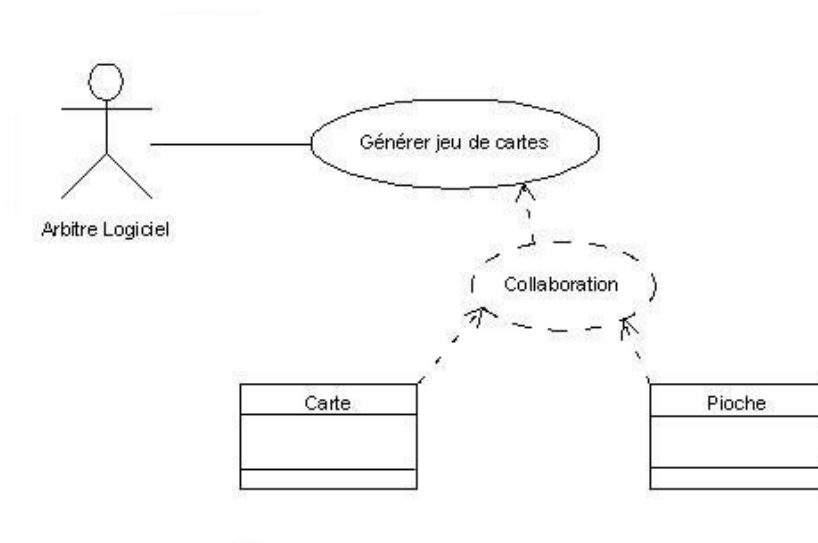


Figure 2 : Diagramme des cas d'utilisation de l'action "générer jeu de cartes"

2. L'action "Déterminé un ordre de passage"

Cette action est réalisée automatiquement et déterminera aléatoirement l'ordre de passage des joueurs durant la partie. Cette action se réalise juste après l'action "générer jeu de cartes". Le diagramme des cas d'utilisation va être présenté.

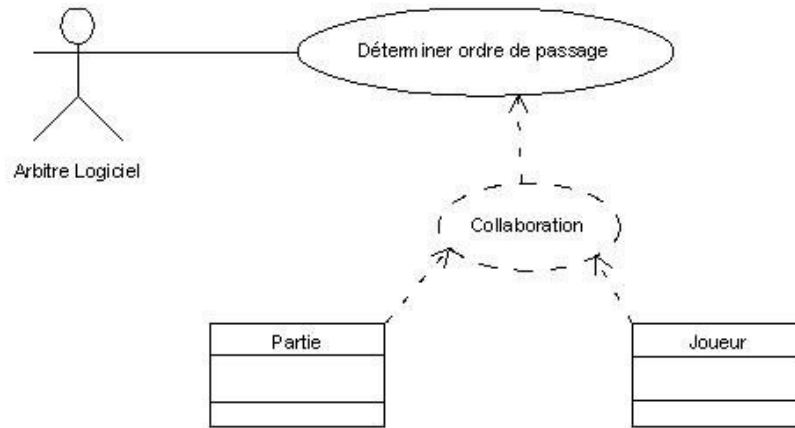


Figure 3 : Diagramme des cas d'utilisation de l'action "Déterminer ordre de passage"

Le diagramme des objets permet de connaître les objets utilisés pour l'action "déterminer ordre de passage". Ces objets seront utilisés lors de l'exécution de l'action.

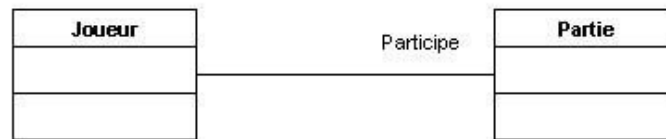


Figure 4 : Diagramme des objets de la figure 2

Le diagramme des classes permet de généraliser les diagrammes précédents concernant l'ordre de passage des joueurs. Ainsi un joueur participe à une partie.

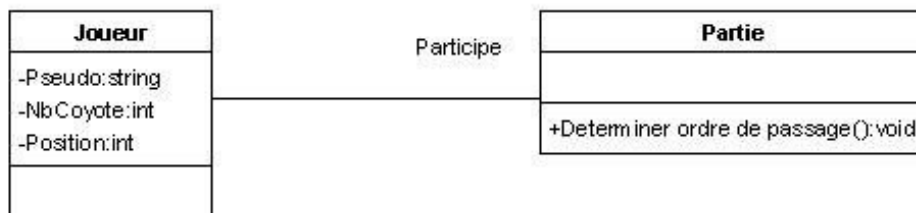


Figure 5 : diagramme des classes

3. L'action "Tirer une carte"

Cette action est exécutée à chaque début de manche. Elle est réalisée pour permettre de distribuer une carte à chaque joueur. Le joueur recevra une carte mais ne pourra pas la voir, elle sera cachée. Cette action se réalise juste après l'action "déterminer ordre de passage". Pour cette action, les diagrammes des cas d'utilisation, des objets et des classes seront établis.

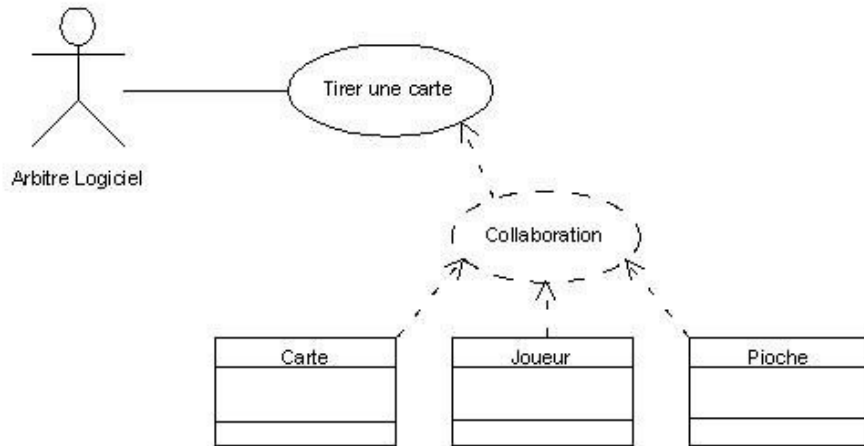


Figure 6 : Diagramme des cas d'utilisation de l'action "Tirer une carte"

Le diagramme des objets permet de déterminer les objets utilisés durant l'exécution de l'action "Tirer une carte". La pioche n'a pas été montrée car c'est une composition de cartes.

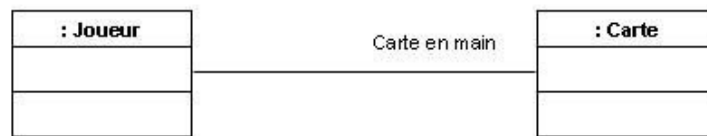


Figure 7 : Diagramme des objets de la figure 3

Le diagramme des classes est une généralisation des ensembles des diagrammes cités précédemment. Un joueur peut donc participer a plusieurs parties. Une partie peut accueillir de 2 à 6 joueurs maximum (ce nombre de joueur pourra être modifié pour permettre d'accueillir plus de 6 joueurs mais par soucis de respect des règles de jeu on a préféré pour l'instant garder le nombre de joueurs autorisés). Durant une partie, un joueur possèdent une et une seule carte qui provient d'une pioche qui est généré au début de la partie.

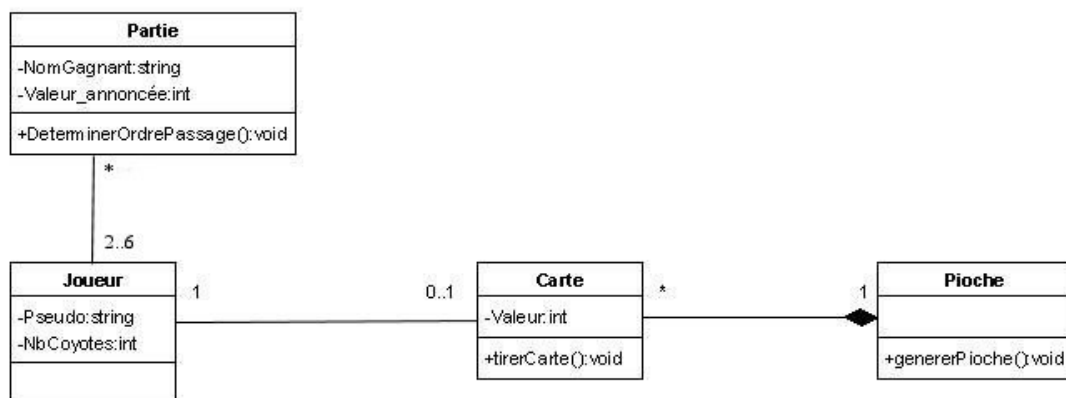


Figure 8 : Extension diagramme des classes figure 5

4. L'action "Envoi des cartes aux autres joueurs"

Cette action permet aux joueurs de voir les cartes de tous les autres joueurs. Les cartes vont être automatiquement envoyées à chaque joueur. Il faut faire attention à ce que la carte du joueur reste face cachée pour ce joueur.

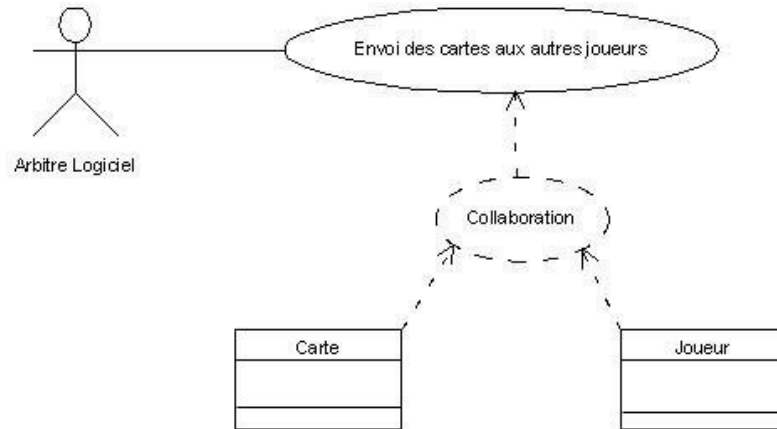


Figure 9 : Diagramme des cas d'utilisation de l'action "Envoi des cartes aux autres joueurs"

5. L'action "Annonce Coyote"

Cette action est réalisée durant le tour du jeu du joueur. Le joueur a le choix : soit il annonce coyote soit il annonce une valeur (ce cas là sera traité plus loin). Si le joueur annonce coyote, la manche se termine et l'action "calcul score total" est exécutée.



Figure 10 : Diagramme des cas d'utilisation de l'action "Annonce coyote"

L'action "calcul score total" permet de faire le calcul de toutes les cartes des joueurs et ainsi permettre de connaître si le joueur ayant annoncé coyote a raison ou pas.

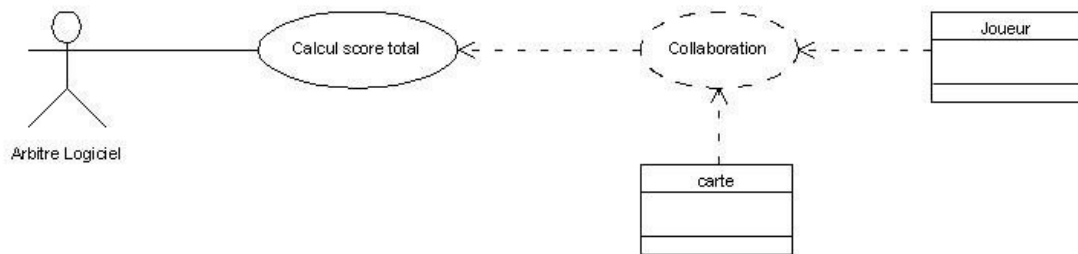


Figure 11 : Diagramme des cas d'utilisation de l'action "Calcul score total"

6. L'action "Annoncer une valeur"

Cette action est réalisée durant le tour de jeu du joueur. Le joueur aura le choix soit d'annoncer "coyote" soit d'annoncer une valeur qui devra être supérieur à celle annoncée par le joueur précédent.

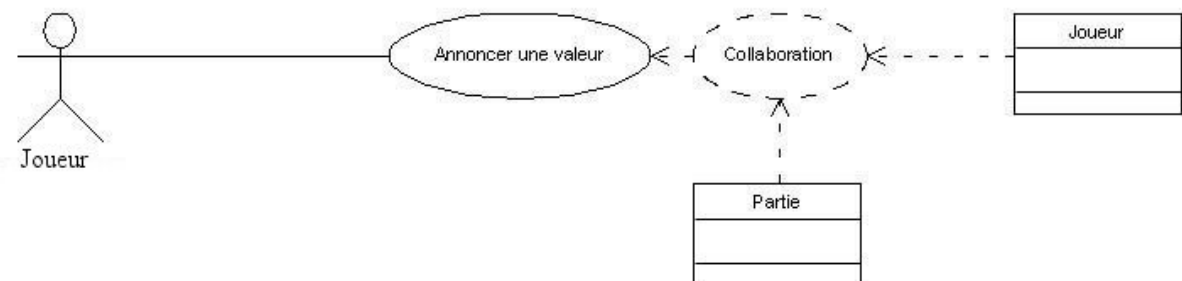


Figure 12 : Diagramme des cas d'utilisation de l'action "Annoncer une valeur"

7. L'action "Envoi d'un coyote"

Cette action intervient juste après l'action "annonce coyote". Cette action va déterminer à qui le coyote va être attribué c'est-à-dire qui a perdu la partie.

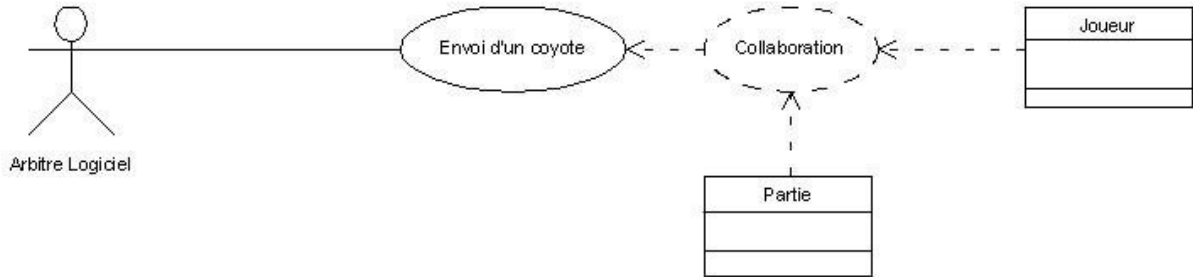


Figure 13 : Diagramme des cas d'utilisation de l'action "Envoi d'un coyote"

8. L'action "Joueur mort"

Cette action est réalisée après l'action "calcul score total". Cette action va permettre de savoir si un joueur est mort c'est-à-dire s'il a atteint un total de 3 coyotes. Cette action correspond à la dernière action d'une manche.

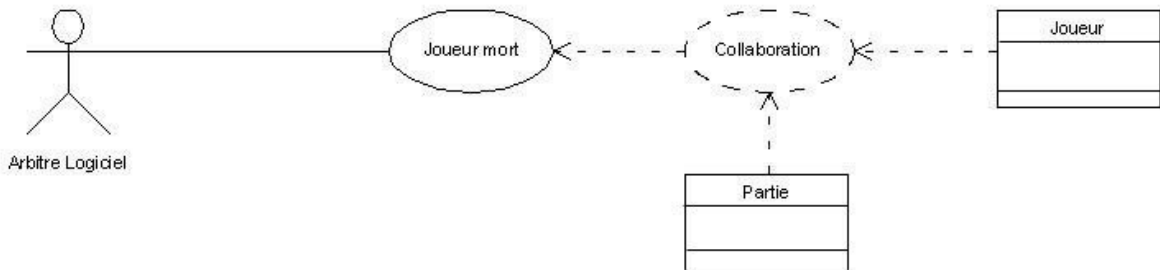


Figure 14 : Diagramme des cas d'utilisation de l'action "joueur mort"

9. L'action "Ajouter carte"

Cette action ne peut être réalisée au cours d'une partie. Cette action correspond à un module du jeu permettant aux différents joueurs de pouvoir ajouter de nouvelles cartes au jeu du coyote.

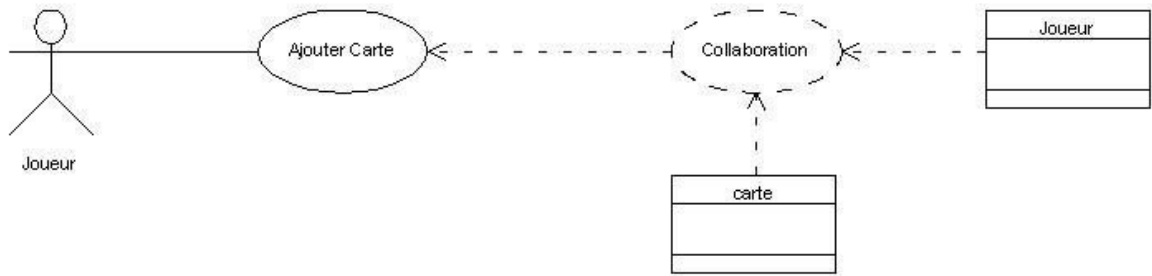


Figure 15 : Diagramme des cas d'utilisation de l'action "Ajouter Carte"

Lorsque l'action sera réalisée, un objet joueur va être créé ainsi qu'un objet Carte. L'objet joueur servira simplement à connaître l'auteur de la carte et l'objet carte permettra de contenir les données de la carte qu'il aura créé. Le diagramme des objets nous permet de voir les objets présents lors de l'exécution de cette action.

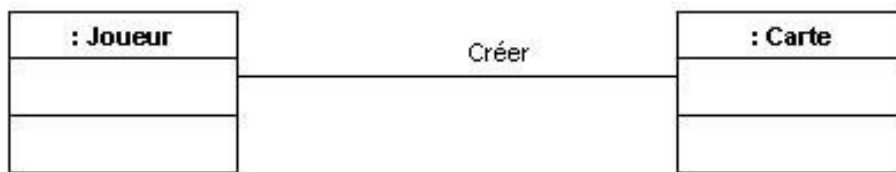


Figure 16 : Diagramme des objets de la figure 16

Le diagramme des classes présenté en figure 18 est une généralisation des diagrammes des figures 16 et 17. Cette généralisation n'est pas complète.

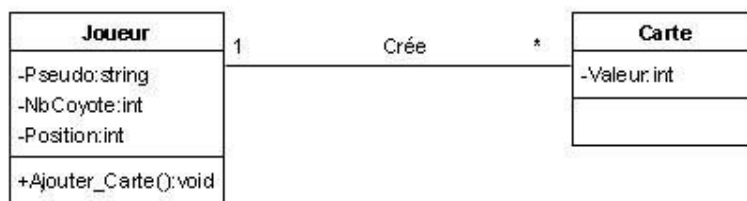


Figure 17 : Diagramme des classes

III Insertion des liens d'héritage

Dans le jeu du coyote en réseau, il existe deux cartes différentes. On peut distinguer la carte qui est valide c'est-à-dire la carte qui va être utilisée dans une pioche et qui sera jouée au cours d'une partie. Et il y a aussi la nouvelle carte qui sera créée par un joueur et qui de ce faite sera en attente de validation avant de pouvoir être jouée.

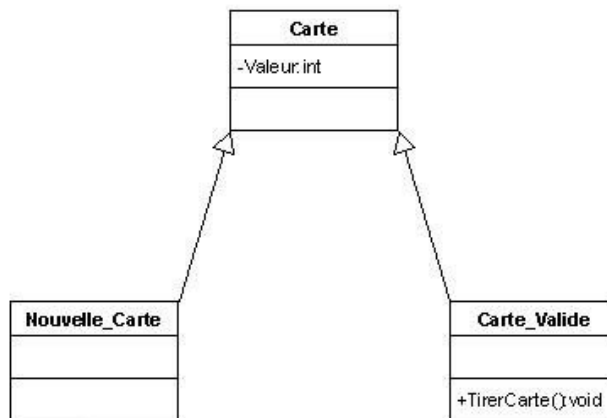


Figure 18 : Modélisation pour les différents types de cartes

IV Diagramme des classes généralisé UML

Le diagramme des classes généralise l'ensemble des diagrammes présentés précédemment. Néanmoins les fonctions d'affichages ont été rajoutées pour permettre au joueur de voir les cartes ainsi que les données de la partie.

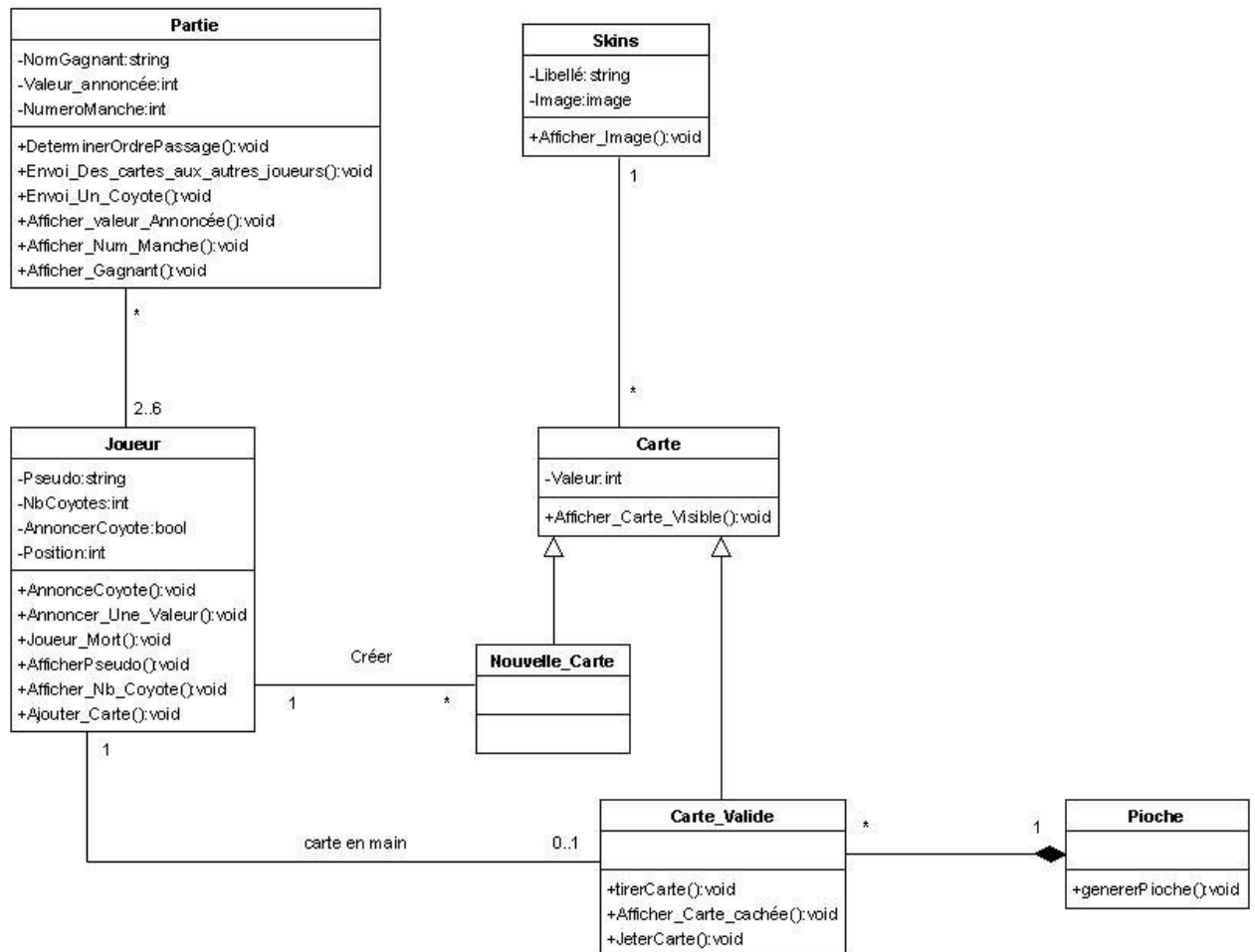


Figure 19 : Extension du diagramme des classes figure 8 avec liens d'héritages et méthode d'affichage

V La persistance des données

Pour permettre de stocker les données des joueurs (statistiques), il nous faut une base de données. Cela va permettre une persistance des données utilisateurs. Cette base de données sera stockée sur le serveur principal.

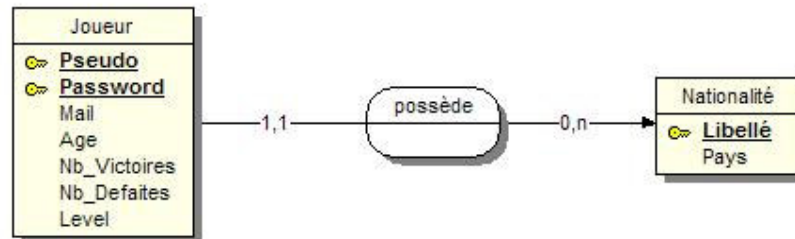


Figure 20 : Modèle conceptuel des données pour la persistance des statistiques des joueurs

En plus des statistiques des joueurs, les cartes vont être stockées. La persistance des cartes sert à permettre d'augmenter le nombre de cartes. Les joueurs vont pouvoir ajouter une ou plusieurs cartes au jeu. Avant que la carte soit acceptée dans le jeu, elle devra être validée par le maître du jeu pour éviter d'avoir trop de cartes ou des cartes trop spéciales.

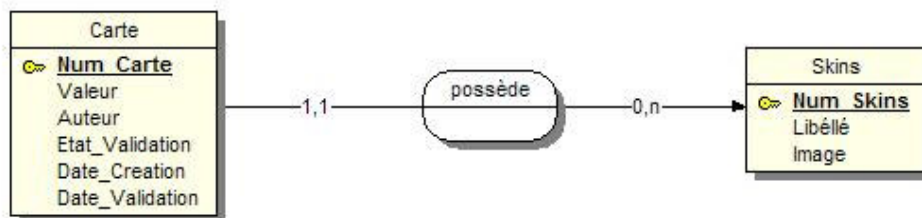


Figure 21 : Modèle conceptuel des données pour la persistance des cartes

Conclusion

Cette étude est certes pas complète mais donne un point de départ à la conception et la réalisation du jeu du coyote en réseau. Pour que cette étude conceptuelle soit plus poussée, la réalisation des diagrammes des activités et des états de transitions serait un plus. Les règles de base du jeu du coyote ont été respectées et quelques fonctionnalités supplémentaires ont été rajoutées.