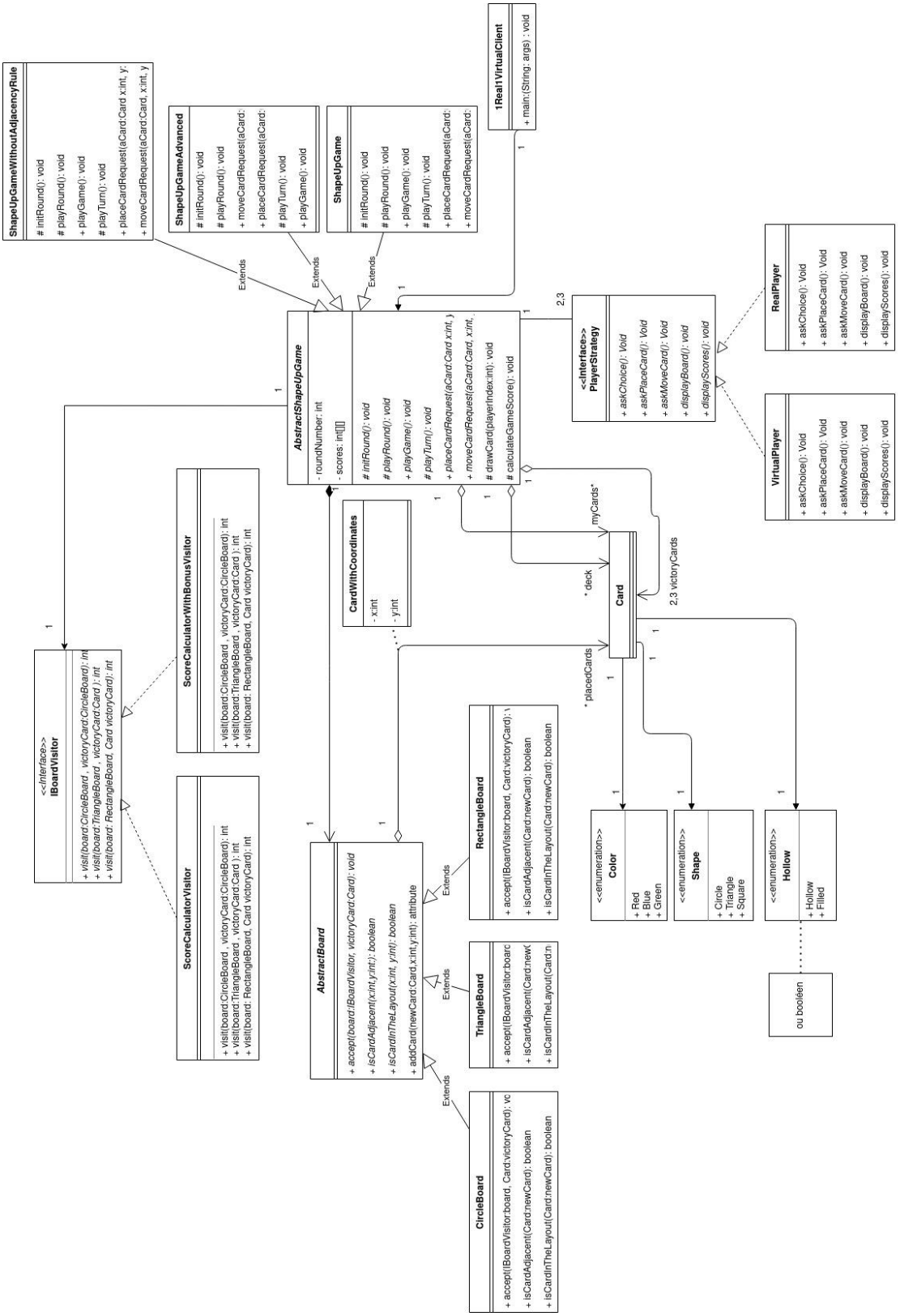
LO02 - Principe et pratique de la programmation orientée objet

Rapport de projet: Shape Up Game

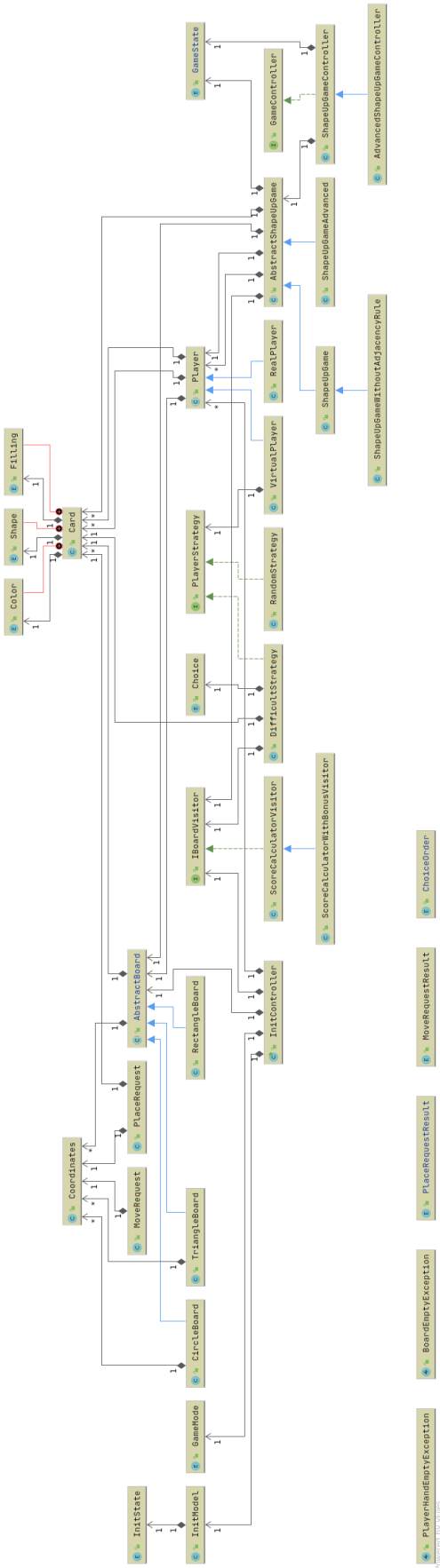
Jacques MIRONNEAU - Baptiste GUICHARD

Diagrammes de classe:

Version initiale



Version actuelle (sans les attributs et méthodes)



Version actuelle (avec les attributs et méthodes)

Changements entre les deux diagrammes:

- **Joueurs et stratégie:**

Par rapport à la version initiale nous avons changé la manière dont nous avions implémenté le joueur virtuel et sa stratégie: en effet nous avions choisi de définir les joueurs comme étant des stratégies. Néanmoins maintenant, nous avons une classe abstraite Player, puis deux implémentations (RealPlayer et VirtualPlayer).
Le virtual player possède un attribut de type PlayerStrategy (qui est une interface implémentée par 2 classes: RandomStrategy et DifficultStrategy).

- **AJout du patron MVC**

L'ajout du patron MVC a amené plusieurs classes supplémentaires: l'interface GameController, ainsi que ses deux implémentations, le InitController, puis de nombreuses classes de vues. (Sans compter la partie IHM, il y a 3 classes supplémentaires).

- **Ajout d'objet de communication**

Nous avons également ajouté des objets de communications, comme le MoveRequest, ou le PlaceRequest, qui sont des objets encapsulant deux coordonnées ou une carte et une coordonnée, afin de représenter un mouvement et un déplacement. Nous avons également ajouté quelques énumérations afin de simplifier le code et d'éviter d'utiliser des entiers ou des chaînes de caractères.

État actuel de l'application

Le jeu tourne avec Java 14 et est jouable de deux à trois joueurs (que ceux-ci soient physiques ou virtuels).
Le joueur virtuel possède deux modes de difficulté (Facile et Difficile) implémentés avec l'utilisation du patron de conception Strategy.

Il existe des variantes sur le mode de jeu: on peut jouer en mode normal, avancé, ou encore dans un 3ème mode de jeu ou la règle d'adjacence n'existe plus.

Le jeu est jouable avec 3 plateaux de formes différentes: rectangle, triangle et cercle.
Les plateaux rectangles et triangles sont orientables dans différents sens au fur et à mesure de la partie.

On peut également choisir la manière dont est calculé le score, nous avons utilisé le patron de conception *Visitor*, et ajouté une variante pour le calcul de celui-ci.
Il est donc possible de choisir la forme du plateau, les règles du jeu et la manière dont le score est calculé. (En plus du choix des joueurs).

Le jeu est jouable en ligne de commande et en interface graphique de manière concurrente: on peut donc jouer un tour en IHM ou en console. On peut également choisir à tout moment de passer d'un mode à un autre durant un tour.
Ceci est possible grâce au patron MVC que nous avons intégré dans le jeu.

Le jeu se joue en glisser-déposer: pour placer une carte il faut glisser une carte (depuis la main du joueur en bas à gauche de l'écran) jusqu'au plateau.
Pour un mouvement, il suffit de faire glisser une carte depuis l'emplacement voulu
Pour l'interface graphique, nous avons choisi de représenter les cartes comme des pierres qui affichent un hologramme. Nous avons également mis en place une animation de "glitch" sur ces hologrammes pour rendre le jeu un peu plus vivant. Le jeu possède également une musique qui est jouée durant toute la partie.

Le jeu est exécutable depuis un terminal afin de jouer de manière concurrente ou simplement en double cliquant sur le jar afin de jouer graphiquement.

Bugs connus:

- Le jeu possède néanmoins quelques bugs d'affichage (un texte qui se décale, ou une représentation graphique en console se décale (à cause des différents threads)
- Lors du premier tour, si on ne pose pas la carte dans le carré blanc, l'écran ralentit
- Lors d'un mouvement sur l'interface graphique, si l'on déplace une carte de l'extrémité gauche ou droite l'affichage des emplacements disponible est décalé
- Les animations sont parfois ralenties lorsque le nombre de carte devient important