1. ***Introduction***

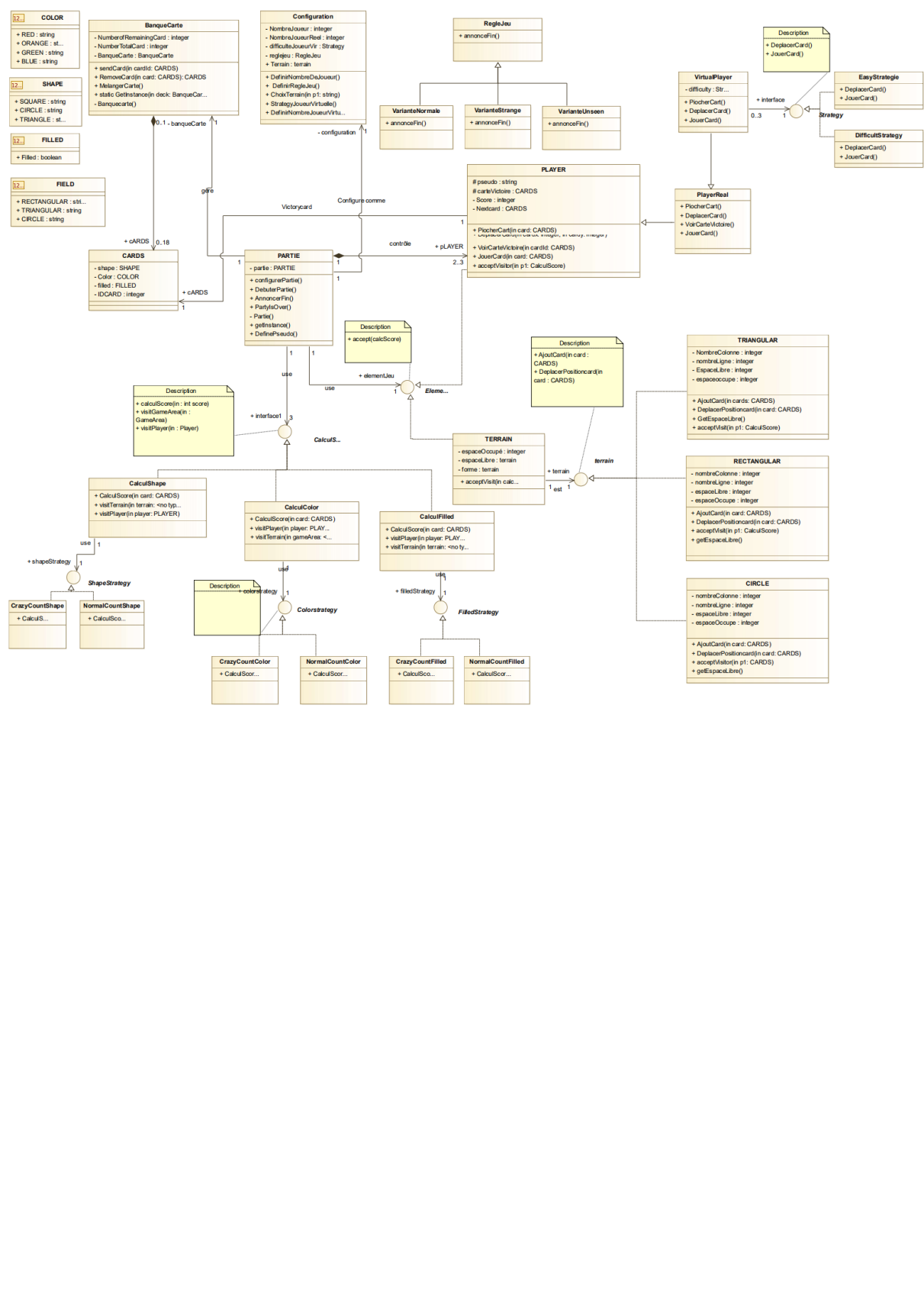
L’unité de valeur intitulée « Principes et pratique de la programmation » objet (code UV : LO02) est un module d’informatique enseignée pendant le semestre d’automne à l’UTT. La problématique de cette UV est de transmettre aux apprenants d’une part des notions de base en modélisation UML et d’autre part des principes fondamentaux et de bonnes pratiques de programmation quand il s’agit du paradigme objet. Ainsi nous avons appris à construire trois types de diagramme UML, à savoir : diagramme de cas d’utilisation, diagramme de classe et diagramme de séquence. Par ailleurs, nous avons appris un panel non exhaustif de patrons de conception de base qui sont très utiles pour produire du code modulaire et réutilisable en programmation orienté objet.

Toutes ces nouvelles notions ont été mises en pratique tout le long du semestre au travers de la réalisation d’un projet de développement logiciel. Ce projet que nous avons décidé d’appeler « Shape-Up » consiste en la modélisation et le développement logiciel d'un jeu dont les règles et les conditions d’utilisation sont spécifiées ici: <http://www.goodlittlegames.co.uk/games/06-shape-up.html>

A ces contraintes fonctionnelles s’ajoutent d’autres contraintes techniques comme l’implémentation obligatoire des patrons visiteur, stratégie et MVC.

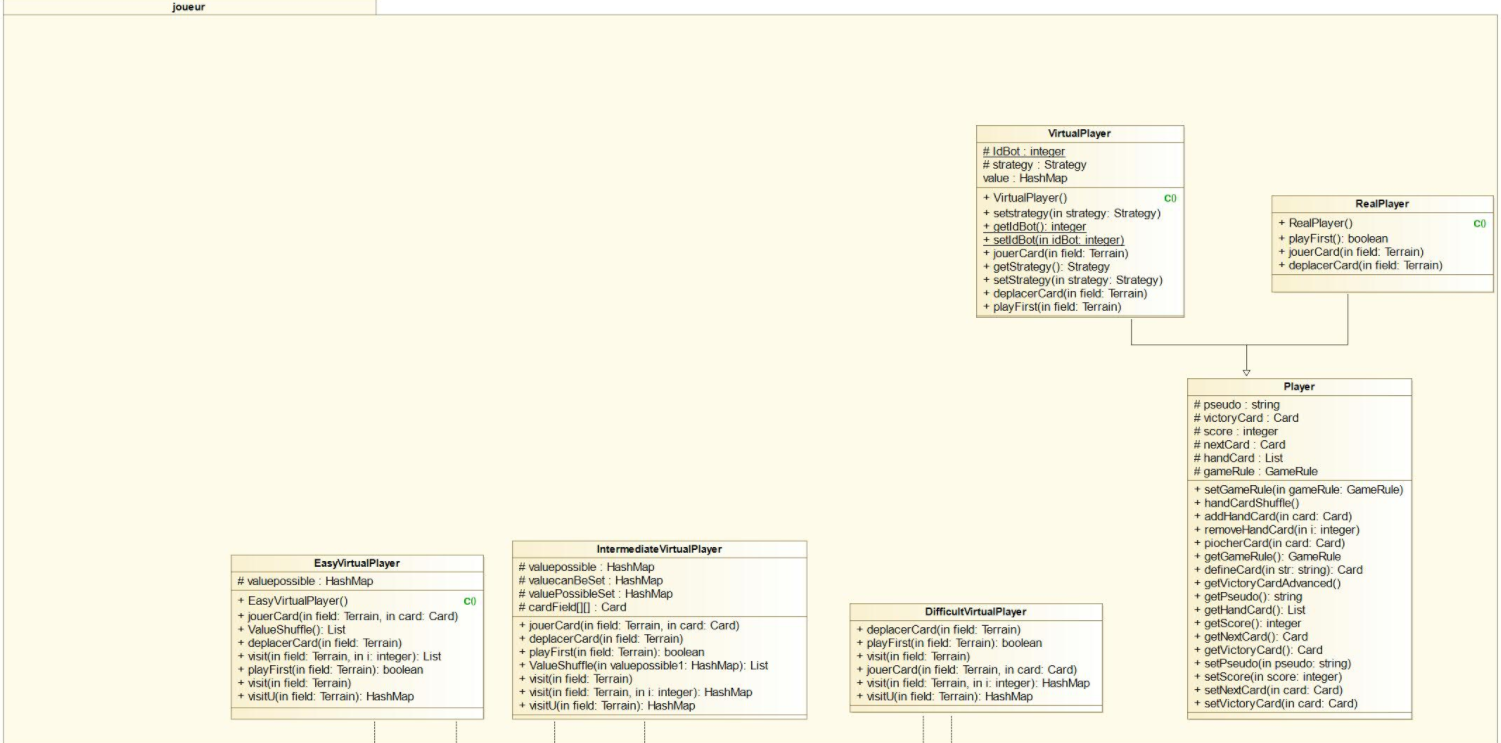
Nous allons donc dans ce document présenter de manière succincte les grands axes du déroulement de ce projet au cours du semestre, c’est-à-dire les Modélisations initiales et finales avant de comparer ces dernières et de donner les enseignements que nous en avons tirés.

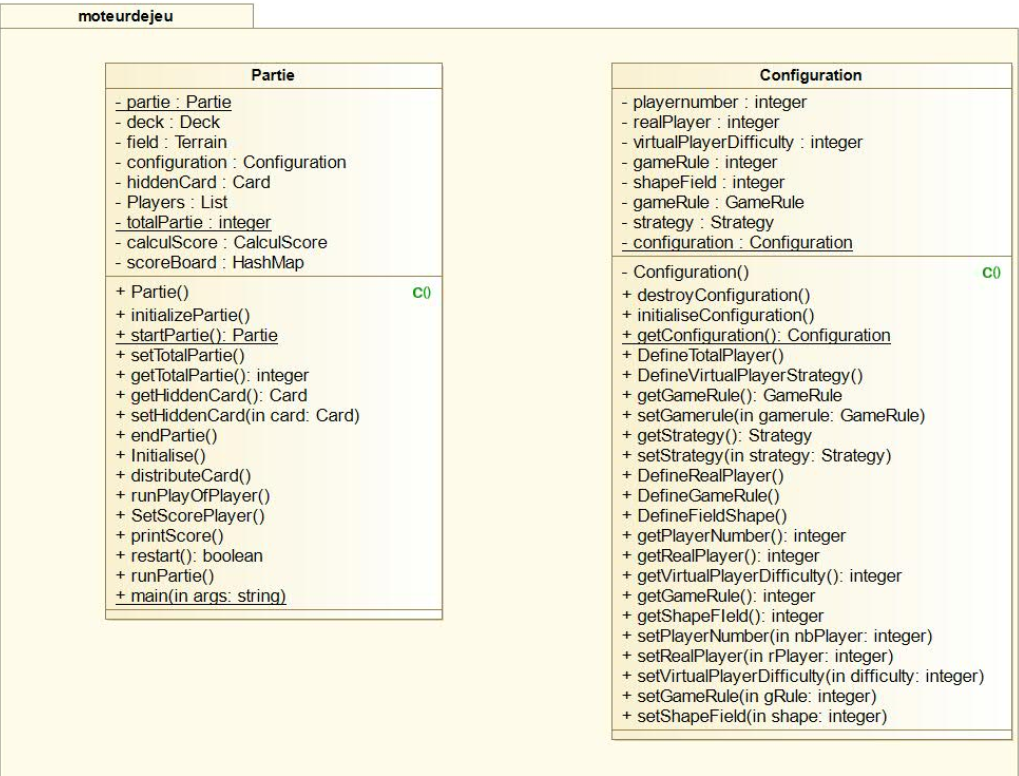
1. ***Diagramme de classe initial***



Cette modélisation a été la ligne directrice du développement du logiciel. Elle ne contient pas toutes les classes qui ont effectivement été implémentées pour rendre le jeu fonctionnel : notamment les classes résultant de l’implémentation du patron MVC (les classes du l’interface graphique et du contrôleur) car ce patron de conception n’avait pas encore été abordé pendant la réalisation de cette modélisation. Toujours dans ce contexte les classes de cette modélisation se sont vues attribuées de nouveaux membres.

1. ***Diagramme de classe actuel***
2. *UML Player*



1. *UML Moteur*
2. *Implémentation du MVC*

Une image contenant texte, carte

Description générée automatiquement

1. *Comparaison*

La version UML actuelles contient les classes de l’interface graphique, les classes des contrôleurs, les classes fonctionnelles telles que les classes permettant de comparer des cartes et des joueurs. Elle contient en plus l’implémentation du patron MVC. De même on peut voir une meilleure implémentation du diagramme de classe car dû au manque réel de pratique lors de la première conception du diagramme classe, nous n’avions pas pu bien implémenté certaines associations.

1. ***Point sur le Jeu***
2. *Bugs*

* Bugs lors du choix de la stratégie de jeu difficile, nous n’avons pas eu le temps de bien arranger le code ;
* Le Terrain en forme triangulaire n’a pas été implémenté ;
* Bug lors du lancement du programme en double cliquant sur le jar, il faut lance le programme en ligne de commande pour éviter le bug ou bien il faut lancer la version du jar qui exécute juste la partie graphique
* Bugs occasionnels lors du passage d’une vue à une autre pendant le jeu lorsque certaines conditions sont réunies ;

1. *Ce qui marche*

* Le Jeu est jouable entièrement dans chacune des vues (graphique et Console), il est aussi jouable sur les deux interfaces à la fois la majeure partie du temps (apparition de quelques bugs)
* Le jeu implémente tous les patrons de conceptions
* Le Jeu admet des variantes dans les règles (normal et avancée)
* Il n’est pas possible pour un Joueur de voir la main d’un autre à moins que celui qui a la main le veuille

1. *Les aspects à améliorer et les axes de développement*

* Corriger les Bugs
* Pouvoir sauver une partie et la relancer
* Terminer d’implémenté la forme triangulaire
* Pouvoir jouer en ligne
* Améliorer le rendu visuel du Jeu