# 

# POO-IG

# Rapport – Projet de POO

Bracchi Pierre

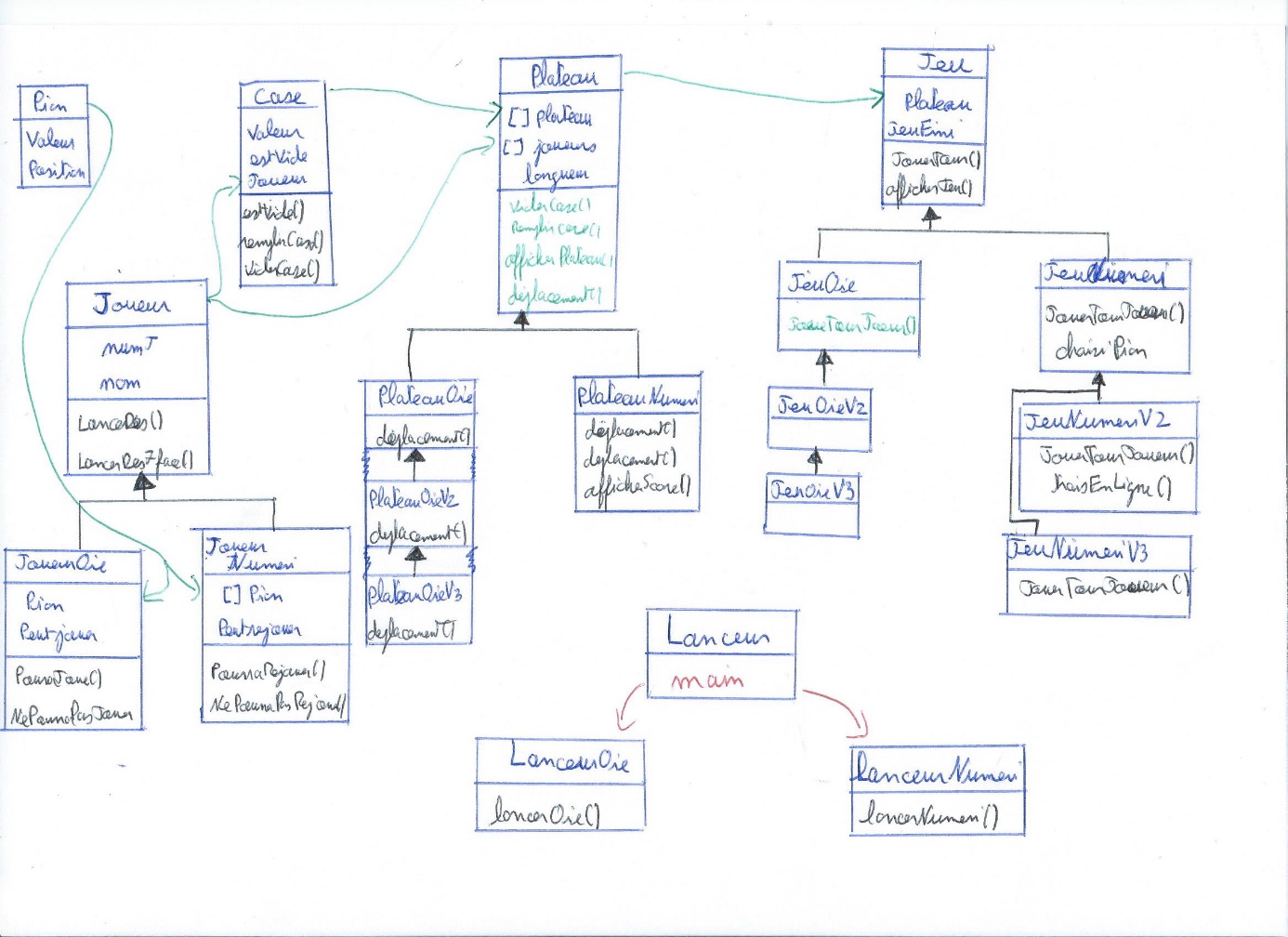
Pour ce projet, je me suis concentré sur la modélisation et la création d’un environnement de développement. J’ai dû laisser la réalisation de l’interface graphique de côtés (essentiellement par manque de temps).

Le but de ce projet étant de faire de la programmation objet et non faire une modélisation à plat, j’ai commencé par décomposer les éléments de chaque jeu et regrouper leur point en commun. Ici, chaque jeu est composé de joueurs, de pions, d’un plateau et se joue avec un ou plusieurs dés.

Afin d’avoir le meilleur niveau d’abstraction possible et une plus grande facilité pour l’implémentation de variantes ou d’autres jeux, j’ai créé des classes relativement larges, voire abstraites en m’inspirant des différents TP vu durant ce semestre.

J’ai alors créé des classes Joueur, Plateau, Jeu, Pion, Case relativement simple ne comportant que des informations générales commun à tous les jeux de plateau. Toutes ces classes possèdent des méthodes de base (toString() , getteur, modification d’attribut, etc…) et un catalogue de méthode que chaque sous classe devra posséder. Par exemple chaque sous classe de la classe Plateau possèdera une méthode déplacement et une méthode afficherPlateau.

Ci-dessous, la représentation graphique de mon modèle :



J’ai implémenté 3 variantes du jeu de l’Oie ainsi que 3 variantes du jeu Numeri.

Pour les jets de dés, j’utilise une méthode static à la classe Joueur. Celle-ci peut être facilement changée en méthode non static.

J’utilise une méthode static isPrasable qui vérifie que mon entrée utilisateur soit bien un int. Elle m’évite à gérer des exceptions. Ayant eu l’idée qu’au milieu de processus, on peut la retrouver dans plusieurs classe. Pour améliorer je pourrais faire une classe a part avec juste cette méthode et l’appeler quand j’en ai besoin. Cela m’épargnera le copier-coller.

Pour le jeu de l’Oie :

Mon implémentation du jeu de l’Oie respecte toutes les règles concernant les cases et jets de dés. L’utilisateur a le choix du nombre de joueur et de la version qu’il souhaite. Il est guidé par le programme pour chaque étape. A chaque tour, il a le choix d’afficher le plateau, jouer un tour et peut quitter le programme à tout moment en tapant ‘exit’ dans le terminal. Quand l’utilisateur choisi de jouer un tour, les jets de dés et déplacements se font automatiquement pour tous les joueurs.

JeuOie correspond à la variante où le jeu se termine quand un pion dépasse la ligne d’arrivée.

JeuOieV2 correspond à la variante où le jeu se termine quand un pion se pose exactement sur la case d’arrivée.

JeuOieV3 correspond à la variante où deux pions ne peuvent cohabiter dans une même case, si c’est le cas, le dernier arrivé ira sur la case libre précédente.

Je n’ai pas eu le temps de faire la variante avec les questions… Mais je pense qu’avec cette modélisation et avec l’aide du TP8 qui est justement sur ce sujet il ne sera pas très compliqué de l’implémenter.

Pour la V1 et la V2 un ensemble de tests et simulations ont été effectués. Tous les problèmes rencontrés ont été corrigés. Ces versions ne devraient plus avoir de bug d’exécution.

La V3 a été soumise à une légère simulation. Un problème est connu et non corrigé : quand le pion recul d’une case il ne prend pas en compte l’effet de cette case.

Pour numeri :

Mon implémentation du jeu Numeri respecte les règles indiquées dans l’énoncé du projet. L’utilisateur a le choix du nombre de joueur et de la version du jeu qu’il souhaite. Il peut quitter le programme à tout moment en tapant ‘exit’ dans le terminal. Ici aussi le jet de dés est automatique, le joueur n’est qu’à choisir le(s) pion(s) qu’il souhaite bouger dans le bon ordre.

A chaque tour le programme affiche l’emplacement des pions de tous les joueurs ainsi que le score actuel.

La V1 respecte les règles de base et le jeu se termine quand les 3 dernières cases sont occupées.

La V2 permet au joueur de rejouer s’il possède 3 pions à lui alignés.

La V3 permet au joueur de faire reculer un pion (ici n’importe lequel s’il est sur le plateau) s’il fait 0 au lancer de dé et rejoue ensuite.

Un problème a été rencontré et non corrigé. Si par exemple la valeur du dé est 4 et que le joueur choisi en premier le pion 2, il se retrouve bloqué. Il ne peut pas choisir une 2e fois le pion 2 et ne peut pas compléter avec d’autres pions pour faire 4. Il faudrait que j’ajoute une fonctionnalité pour annuler le pion choisi pour éviter cette erreur.

Les possibilités de l’utilisateur, comme changer son nom par exemple, n’ont pas été approfondies. Sachant que j’avais toujours l’optique de faire l’interface graphique. De plus, je ne pense pas que c’était le but principal de ce projet.