

## **Suivi projet : Morpion solitaire**

### **Version 0 :**

Dans le cadre du cours de java avancée et du projet de Morpion Solitaire. Un premier diagramme de classe à été conceptualisé :

- utilisation du patron Modèle Vue Contrôleur
- choix de l'utilisation de Scene Builder pour les vues
- package séparé pour le modèle et l'implémentation des algorithmes de recherche
- choix du patron Factory pour la gestion d'algorithmes multiples
- différenciation du Joueur et des Sauvegarde
- classe Utils regroupant des méthodes communes à Grille5T et Grille5D avec une classe Grille abstraite

### **Version 1 :**

A la suite de la première version, le choix de séparer les jalons d'implémentation en deux parties a été fait. Dans un premier temps il est nécessaire d'avoir un moteur de jeux fonctionnel, avec des classes Vue et Contrôleur réduites (et en Java) de manière à faire tourner le jeu en console avant de poursuivre vers une implémentation des packages Vue et Contrôleur en JavaFX

### **Version 2 :**

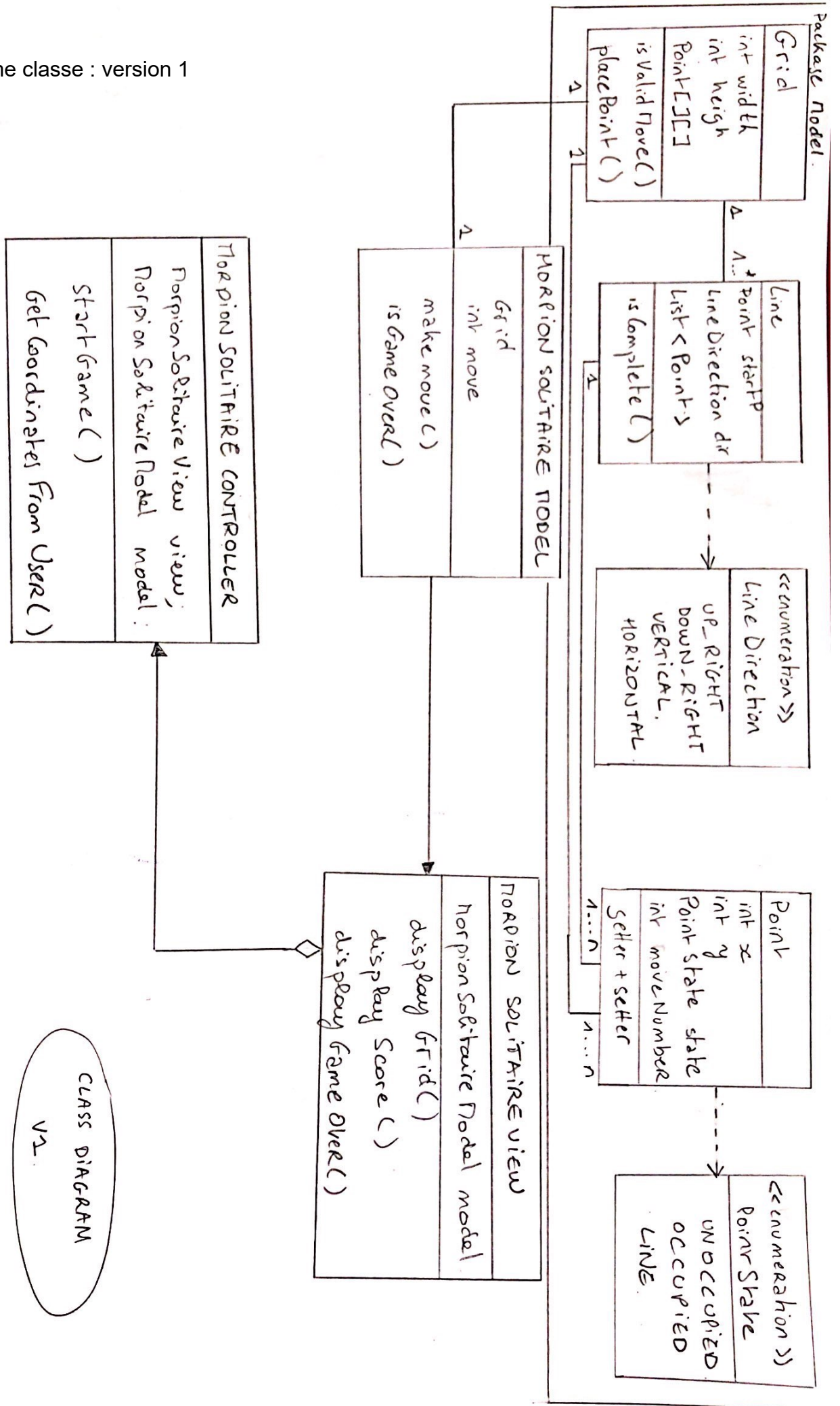
Une fois l'application fonctionnelle en console nous pouvons implémenter, deux grilles différentes :

- Grid transformer en interface avec deux nouvelles classes Grid5T et Grid5D
- Implémentation de Joueur.java qui gèrera les sauvegardes et informations relatives à l'utilisateur (score, points, parties jouées, reprendre une partie non finies).
- Ajout d'un package AlgoReserach avec les différents algorithmes de recherche au sein d'une Factory
- Ajout du package de test pour les tests unitaires

### **Version 3 :**

La version suivante, et finale, consistera à implémenter les packages Vue et Contrôleur en JavaFX, qui remplaceront les classes MorpionSolitaireView.java et MorpionSolitaireController.java actuelle pour avoir une interface graphique fonctionnelle.





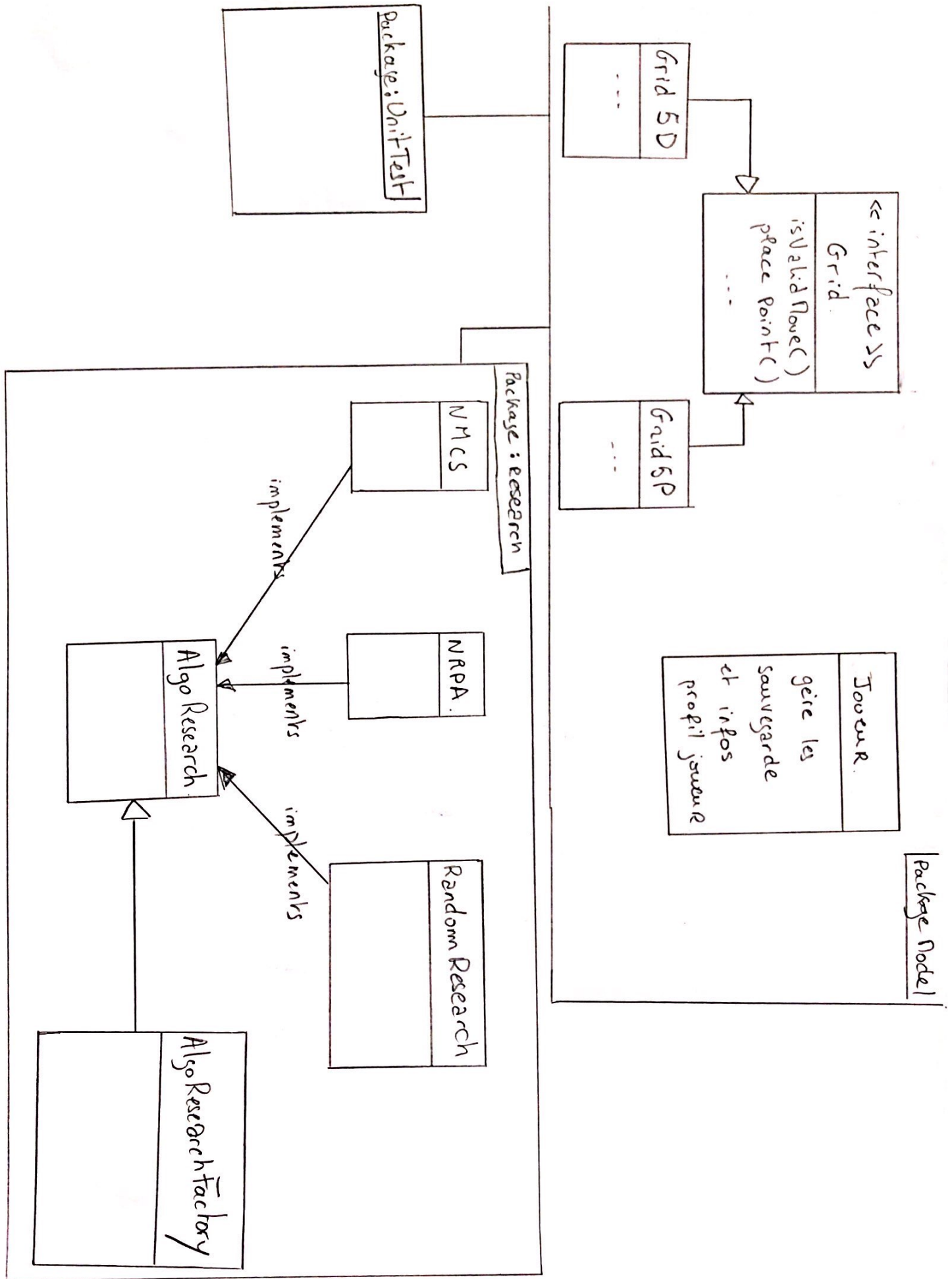


Diagramme classe : version 2