



**IUT de Paris - Rives de Seine**  
Université Paris Cité



---

# PROJET HEX

---

Développer en Java



25 DECEMBRE 2022

OUSTI Redouane 202, COLLOMBET Nathan 201, ZHANG Anxian 201, LIN Xingtong 201

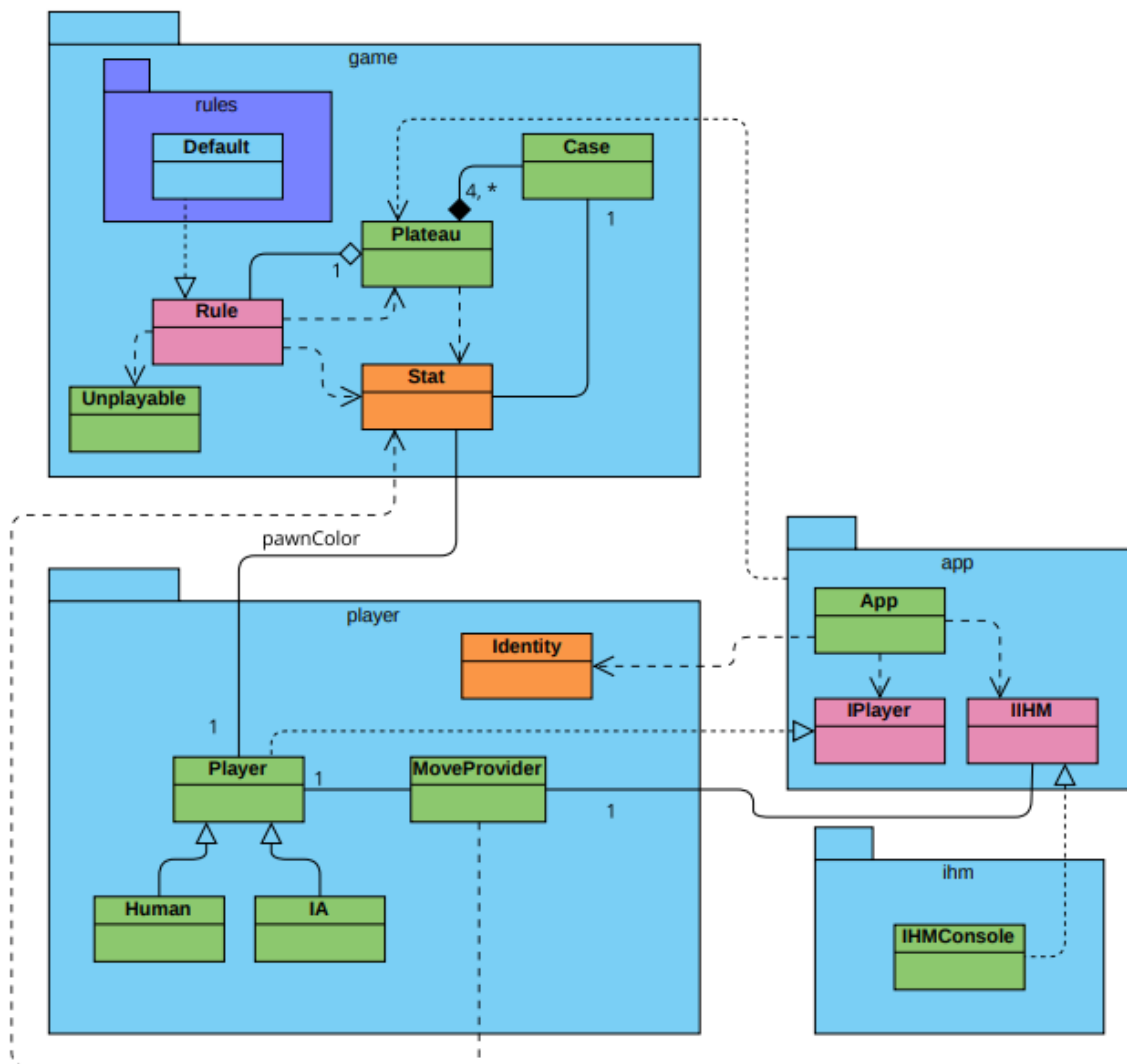
## Table des matières

<b>Introduction .....</b>	<b>2</b>
<b>Diagramme d'architecture .....</b>	<b>2</b>
<b>Tests effectués .....</b>	<b>3</b>
Fichier PlateauTest.....	3
Fichier PlayerTest.....	3
<b>Etapes pour l'ajouts de nouvelles fonctionnalités.....</b>	<b>3</b>
S'il y a modifications/ajouts de règles.....	3
S'il y a modifications/ajouts des joueurs simulés .....	3
<b>Bilan du projet.....</b>	<b>4</b>
Les difficultés rencontrées .....	4
Ce qui est réussi .....	4
Ce qui peut être amélioré.....	5

## I. Introduction

Nous avons réussi à réaliser toutes les fonctionnalités du jeu, tout est fonctionnel (tests et l'application). Cependant nous ne sommes pas certains d'avoir appliqué correctement les principes SOLID (notamment SRP).

## II. Diagramme d'architecture



### III. Tests effectués

*PS : toutes les méthodes qui seront citées sont de visibilité publique*

❖ **Fichier PlateauTest (test de la classe Plateau) les fonctionnalités testées sont :**

- int taille() ,
- int nbOfUsableCase() ,
- boolean isFull(),
- String toString(),
- void play(),
- exception Unplayable (en Junit 5),
- Stat winner().

❖ **Fichier PlayerTest (test de la classe Player) la fonctionnalité testée est :**

- Stat getPawnColor().

### IV. Etapes pour l'ajouts de nouvelles fonctionnalités

❖ **S'il y a modifications/ajouts de règles :**

- Créer la nouvelle classe,
- Implémenter L'interface Rule,
- Modifier l'instance de l'ancienne règle dans Plateau.

❖ **S'il y a modifications/ajouts des joueurs simulés :**

- Créer la nouvelle classe,
- L'étendre avec la classe Player,
- Changer les anciennes instances des joueurs simulés qui se trouvent dans la classe App par la nouvelle.

## V. Bilan du projet

### ❖ Les difficultés rencontrées

#### La Conception :

Au début du projet nous avons mis au point une première version d'un digramme d'architecture possible. Mais nous nous sommes trop investis pour finalement tout changé 2 semaines avant le rendu. C'est donc à ce moment que nous nous sommes rendu compte qu'une bonne conception est très importante.

#### GitHub :

Au début du projet certains membres étaient assez réticents à l'utilisation de cet outil qui ne leur semblait pas très naturelle. Nous utilisions donc un autre outil de code collaboratif (Replit). Nous nous sommes par la suite résignés à utiliser GitHub après un peu d'acclimatation à l'outil. Un seul membre a donc push toutes les avancées du projet jusque-là.

#### La recherche de bug :

Nous avons passé beaucoup de temps sur le développement de la fonction récursive *winner* (qui permet de connaître le joueur gagnant). En effet nous nous sommes confrontés à de nombreuses situations où la méthode n'effectuait pas ce que nous avons mis théoriquement en place. Et les moments les plus frustrants et satisfaisants étaient lorsque nous nous rendons compte que le problème c'était une ligne de code qui était placée au mauvais endroit et nous mettions plus d'une heure à le trouver.

### ❖ Ce qui est réussi

#### L'organisation :

Lorsqu'une personne fait des modifications ou des ajouts celle-ci informe les autres pour être sûr que tout le monde les prenne en considération. De plus nous avons mis au point un Trello afin de gérer au mieux l'organisation du projet et les tâches qui sont effectuées ou qui doivent l'être.

#### La validation :

Durant tout le cycle de développement, à chaque fois que nous finissions une partie ou une méthode importante, nous le testions tout de suite pour nous assurer de son bon fonctionnement. Les tests effectués étaient possibles grâce aux méthodes `toString`. Bien sûr, nous n'excluons pas les `System.out` qui nous ont permis de visualiser ce qui était stocké dans une ou plusieurs variable(s). Si le(s) résultat(s) est/sont cohérent(s) nous validions la méthode puis passions au codage de la suivante. Quand nous finissions une partie, nous testions la où les classe(s) en testant divers cas qui nous viennent.

### ❖ Ce qui peut être amélioré

#### La Conception :

Nous pensions qu'il faudrait que nous revoyions plus en détails comment réaliser une conception efficace dès le départ afin de ne plus commettre les mêmes erreurs et de perdre du temps dessus. Mais aussi la bonne application des principes SOLID (nous avons notamment eu un TP qui a sauté (groupe 201) un lundi matin à cause de la cyber attaque).