

# Théorie des jeux

## Rapport de projet

### Jeu du troll

Hedwin Bonnavaud  
hedwin.bonnavaud@etu.univ-lyon1.fr

Tristan Guillard  
tristan.guillard@etu.univ-lyon1.fr

UCBL Lyon 1 - Rapport TDJ

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Implémentation</b>	<b>2</b>
2.1	Modèle, Vue, Controller . . . . .	2
2.2	Stratégie prudente . . . . .	2
2.3	Tournois . . . . .	2
<b>3</b>	<b>Résultats</b>	<b>3</b>

## 1 Introduction

Le jeu que nous avons étudié pour ce projet suit les règles suivantes

- 2 joueur.euse.s s'affrontent sur un plateau avec un nombre de cases impaires.
- Les joueur.euse.s commencent la partie avec le même nombre de pierres.
- À chaque tour, les 2 joueur.euse.s choisissent un nombre de pierres à lancer :
  - Les joueur.euse.s perdent le nombre de pierres qu'ils ont choisi.
  - Si l'un.e d'elles a lancé plus de pierres que l'autre, le troll s'éloigne d'ellui d'une case.
- Le jeu s'arrête quand l'un.e des deux joueur.euse.s n'a plus de pierres, ou que le troll a atteint l'un.e des deux joueur.euse.s.
- Si le jeu s'arrête car un.e des deux joueur.euse.s n'a plus de pierres, le troll avance dans sa direction du nombre de pierres restant à l'autre joueur.euse.
- Quand le jeu s'arrête, celui dont le troll est le plus proche perd la partie, avec égalité si le troll se trouve au milieu du plateau.

Nous avons eu pour but de coder ce jeu et des stratégies, notamment la stratégie prudente.

## 2 Implémentation

### 2.1 Modèle, Vue, Controller

Nous avons mis en place le pattern MVC décrit dans la figure 1. La figure a été légèrement simplifiée par rapport au code. Les fichiers dans `fichier lançables` sont les fichiers exécutables pour le projet.

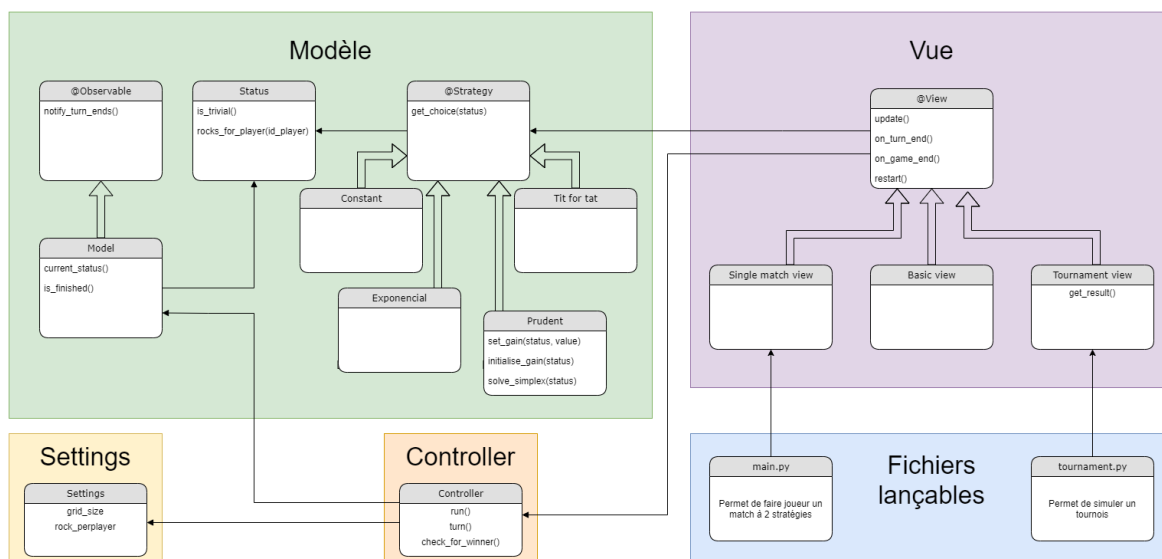


FIGURE 1 – Diagramme UML du MVC de notre projet

### 2.2 Stratégie prudente

Nous avons implémenté la stratégie prudente en résolvant un simplex grâce à la fonction `linprog` de la bibliothèque `scipy.optimize`. Le simplex est entièrement résolu pour tous les cas possibles (déterminés par les `Settings`) au début du code, puis la stratégie prudente va chercher dans le tableau à 3 dimensions quel probabilité de coup jouer à chaque tour.

### 2.3 Tournois

Nous avons développé un système de tournois, où chaque stratégie va affronter toutes les autres sur un certain nombre de parties (10000 par exemple). On affiche ensuite les résultats dans un tableau avec un score par match allant de  $-1$  à  $1$ ,  $1$  étant lorsque la stratégie "player1" a gagné tous ses matchs, et  $1$  lorsque la stratégie "player2" a gagné tous les siens, en passant par  $0$  qui signifie nombre de matchs gagné égaux (cf figure 2 dans la section 3 pour voir un résultat)

### 3 Résultats

```
=====
===      Welcome to the bit troll game tournament      ===
=====
Tournament initialisation ...
Take a look at these rules :
- Games per match: 10000

And here are our candidates !!
- Prudent strategy
- Exponential strategy with lambda 1.0
- Exponential strategy with lambda 1.5
- Tit for tat strategy
- Constant strategy with value 2
- Constant strategy with value 5

Tournament is done ! Here are the results:
- Players 1 candidates :
  - 1 : Prudent strategy
  - 2 : Exponential strategy with lambda 1.0
  - 3 : Exponential strategy with lambda 1.5
  - 4 : Tit for tat strategy
  - 5 : Constant strategy with value 2
  - 6 : Constant strategy with value 5
- Players 2 candidates :
  - 1 : Prudent strategy
  - 2 : Exponential strategy with lambda 1.0
  - 3 : Exponential strategy with lambda 1.5
  - 4 : Tit for tat strategy
  - 5 : Constant strategy with value 2
  - 6 : Constant strategy with value 5

In this matrix you can find player 1 gain value.

Players 1 id are in the left column, and player 2 id are in the first line
      [' 1 ', ' 2 ', ' 3 ', ' 4 ', ' 5 ', ' 6 ' ]
-----
- 1 : ['0.0', '0.3', '0.5', '0.5', '-0.0', '0.1']
- 2 : ['-0.3', '-0.0', '0.0', '-0.0', '-0.9', '-0.9']
- 3 : ['-0.5', '-0.0', '0.0', '-0.1', '-0.8', '-0.6']
- 4 : ['-0.5', '0.0', '0.1', '0.0', '0.0', '0.0']
- 5 : ['-0.0', '0.9', '0.8', '0.0', '0.0', '-1.0']
- 6 : ['-0.1', '0.9', '0.6', '-0.0', '1.0', '0.0']
```

FIGURE 2 – Résultat d'un tournois

On voit bien que les stratégies prudentes ne perdent jamais leur matchs (bien qu'elles perdent certaines parties bien sûr). On voit aussi que l'issue des matchs des stratégies non mixtes est déterministe (une stratégie gagnera toujours contre l'autre). Enfin on vérifie que chaque stratégie ne gagne ni ne perd jamais contre elle même (bien qu'il n'y ai pas forcément que des matchs nuls).