

# Game theory projet: strategies for the game Troll&Castles

Guilhem MARION, Robert KALNA

Mai 2018

## 1 Strategies

### 1.1 Random number of stones

```
def strategy_random(game, previous_parties):
    number_of_stones_of_enemy = min(game.stockGauche, game.stockDroite + 1)
    return int(np.random.choice(range(1, number_of_stones_of_enemy + 1)))
```

### 1.2 Always throw 2 stones

```
def strategy_always_throw_two(game, previous_parties):
    number_of_stones = game.stockGauche
    return min(2, number_of_stones)
```

### 1.3 Gaussian with location of 2 and variance of 0.5

```
def strategy_gaussian(game, previous_parties):
    stones_to_throw = np.random.normal(2, 0.5)
    if stones_to_throw > game.stockGauche:
        stones_to_throw = min(np.random.normal(game.stockGauche//2, 3), game.stockGauche)
    return int(max(stones_to_throw, 1))
```

### 1.4 Strategy leading to Nash equilibrium

Distributions have been calculated before and stored in pickles: `distributions` is an object loaded from a pickle. There is one for games with 7 fields and another with 15 fields. The tables of the utilities have also been calculated and stored in pickles, see `field7/utilities.pkl` and `field15/utilities.pkl`.

```
def strategy_of_nash(game, previous_parties):
    troll_position = int(game.positionTroll - (game.nombreCases - 1) // 2)
    stones_left = game.stockGauche
    stones_right = game.stockDroite
    if (stones_left, stones_right, troll_position) in distributions:
        ((distribution, distribution_ind), g) = distributions[
            stones_left, stones_right, troll_position]
    else:
        ((distribution, distribution_ind), g) = db.calculate_what_to_play(
            stones_left, stones_right, troll_position)
    distribution = np.array(distribution)
    distribution /= distribution.sum()
    X = np.random.choice(distribution_ind, 1, p=distribution)
    return int(X[0])
```

## 1.5 Eager version of the strategy leading to Nash equilibrium

```
def strategy_nash_eager(game, previous_parties):
    troll_position = int(game.positionTroll - (game.nombreCases - 1) // 2)
    stones_left = game.stockGauche
    stones_right = game.stockDroite
    if (stones_left, stones_right, troll_position) in distributions:
        ((distribution, distribution_ind), g) = distributions[
            stones_left, stones_right, troll_position]
    else:
        ((distribution, distribution_ind), g) = db.calculate_what_to_play(
            stones_left, stones_right, troll_position)
    return int(np.array(distribution).argmax() + 1)
```

## 2 Number of fields: 7, stones: 15

### 2.1 Strategy of nash VS random number of stones

----- Resultats de la simulation -----

Matches prevus : 1000

Matches joues : 1000

Victoires du joueur de gauche : 951

Victoires du joueur de droite : 42

Matches nuls : 7

Victoire du joueur de gauche !

### 2.2 Strategy of nash VS eager version of strategy of nash

----- Resultats de la simulation -----

Matches prevus : 1000

Matches joues : 1000

Victoires du joueur de gauche : 275

Victoires du joueur de droite : 283

Matches nuls : 442

Victoire du joueur de droite !

### 2.3 Strategy of nash VS gaussian with location of 2 and variance of 0.5

----- Resultats de la simulation -----

Matches prevus : 1000

Matches joues : 1000

Victoires du joueur de gauche : 675

Victoires du joueur de droite : 275

Matches nuls : 50

Victoire du joueur de gauche !

## 2.4 Strategy of nash VS always throw two stones

----- Resultats de la simulation -----

Matches prevus : 1000

Matches joues : 1000

Victoires du joueur de gauche : 342

Victoires du joueur de droite : 378

Matches nuls : 280

Victoire du joueur de droite !

## 2.5 Nash equilibrium

----- Resultats de la simulation -----

Matches prevus : 1000

Matches joues : 1000

Victoires du joueur de gauche : 309

Victoires du joueur de droite : 324

Matches nuls : 367

Victoire du joueur de droite !

## 3 Number of fields: 7, stones: 30

### 3.1 Strategy of nash VS random number of stones

----- Resultats de la simulation -----

Matches prevus : 1000

Matches joues : 1000

Victoires du joueur de gauche : 947

Victoires du joueur de droite : 50

Matches nuls : 3

Victoire du joueur de gauche !

### 3.2 Strategy of nash VS eager version of strategy of nash

----- Resultats de la simulation -----

Matches prevus : 1000

Matches joues : 1000

Victoires du joueur de gauche : 438

Victoires du joueur de droite : 457

Matches nuls : 105

Victoire du joueur de droite !

### 3.3 Strategy of nash VS gaussian with location of 2 and variance of 0.5

----- Resultats de la simulation -----

Matches prevus : 1000

Matches joues : 1000

Victoires du joueur de gauche : 732

Victoires du joueur de droite : 268

Matches nuls : 0

Victoire du joueur de gauche !

### 3.4 Strategy of nash VS always throw two stones

----- Resultats de la simulation -----

Matches prevus : 1000

Matches joues : 1000

Victoires du joueur de gauche : 617

Victoires du joueur de droite : 383

Matches nuls : 0

Victoire du joueur de gauche !

### 3.5 Nash equilibrium

----- Resultats de la simulation -----

Matches prevus : 1000

Matches joues : 1000

Victoires du joueur de gauche : 451

Victoires du joueur de droite : 491

Matches nuls : 58

Victoire du joueur de droite !

## 4 Number of fields: 15, stones: 30

### 4.1 Strategy of nash VS random number of stones

----- Resultats de la simulation -----

Matches prevus : 1000

Matches joues : 1000

Victoires du joueur de gauche : 1000

Victoires du joueur de droite : 0

Matches nuls : 0

Victoire du joueur de gauche !

## 4.2 Strategy of nash VS eager version of strategy of nash

----- Resultats de la simulation -----

Matches prevus : 1000

Matches joues : 1000

Victoires du joueur de gauche : 0

Victoires du joueur de droite : 1

Matches nuls : 999

Victoire du joueur de droite !

## 4.3 Strategy of nash VS gaussian with location of 2 and variance of 0.5

----- Resultats de la simulation -----

Matches prevus : 1000

Matches joues : 1000

Victoires du joueur de gauche : 476

Victoires du joueur de droite : 214

Matches nuls : 310

Victoire du joueur de gauche !

## 4.4 Strategy of nash VS always throw two stones

----- Resultats de la simulation -----

Matches prevus : 1000

Matches joues : 1000

Victoires du joueur de gauche : 1

Victoires du joueur de droite : 0

Matches nuls : 999

Victoire du joueur de gauche !

## 4.5 Nash equilibrium

----- Resultats de la simulation -----

Matches prevus : 1000

Matches joues : 1000

Victoires du joueur de gauche : 4

Victoires du joueur de droite : 2

Matches nuls : 994

Victoire du joueur de gauche !

## 5 Number of fields: 15, stones: 50

### 5.1 Strategy of nash VS random number of stones

----- Resultats de la simulation -----

Matches prevus : 1000

Matches joues : 1000

Victoires du joueur de gauche : 1000

Victoires du joueur de droite : 0

Matches nuls : 0

Victoire du joueur de gauche !

### 5.2 Strategy of nash VS eager version of strategy of nash

----- Resultats de la simulation -----

Matches prevus : 1000

Matches joues : 1000

Victoires du joueur de gauche : 303

Victoires du joueur de droite : 243

Matches nuls : 454

Victoire du joueur de gauche !

### 5.3 Strategy of nash VS gaussian with location of 2 and variance of 0.5

----- Resultats de la simulation -----

Matches prevus : 1000

Matches joues : 1000

Victoires du joueur de gauche : 734

Victoires du joueur de droite : 254

Matches nuls : 12

Victoire du joueur de gauche !

### 5.4 Strategy of nash VS always throw two stones

----- Resultats de la simulation -----

Matches prevus : 1000

Matches joues : 1000

Victoires du joueur de gauche : 177

Victoires du joueur de droite : 65

Matches nuls : 758

Victoire du joueur de gauche !

### 5.5 Nash equilibrium

----- Resultats de la simulation -----

Matches prévus : 1000

Matches joués : 1000

Victoires du joueur de gauche : 330

Victoires du joueur de droite : 355

Matches nuls : 315

Victoire du joueur de droite !