

# Projet d'UML Dilemme Itéré des Prisonniers

D'après Laurent AUDIBERT

(IUT, département informatique, 1<sup>re</sup> année)

L'objectif du projet consiste à proposer un modèle UML d'une petite application permettant de mettre en œuvre des tournois de stratégies jouant au *Dilemme Itéré des Prisonniers*, puis de proposer une implémentation en Java de cette application.

## 1 Le contexte : du dilemme du prisonnier à l'interaction de tribus

### 1.1 Le dilemme du prisonnier

Deux suspects porteurs d'armes ont été arrêtés devant une banque et mis dans deux cellules séparées. Les deux prévenus ne peuvent pas communiquer et doivent choisir entre avouer qu'ils s'apprêtaient à commettre un hold-up ou ne rien avouer. Les règles que le juge leur impose sont les suivantes :

- si l'un avoue et pas l'autre, celui qui avoue sera libéré en remerciement de sa collaboration et l'autre sera condamné à cinq ans de prison ;
- si aucun n'avoue, ils ne seront condamnés qu'à deux ans de prison, pour port d'arme illégal ;
- et si les deux avouent, ils iront chacun faire quatre ans de prison.

Dans cette situation, il est clair que si les deux s'entendent (pas d'aveu), ils s'en tireront globalement mieux (2 x 3 ans de remise de peine) que si l'un des deux dénonce l'autre (1 x 5 ans de remise de peine). Mais alors l'un peut être tenté de s'en tirer encore mieux en dénonçant son complice. Craignant cela, l'autre risque aussi de dénoncer son complice pour ne pas être le dindon de la farce. Le dilemme est donc : faut-il accepter de couvrir son complice (donc de coopérer avec lui) ou le trahir ?

Ce modèle très simple de la théorie des jeux semble appréhender en miniature les tensions entre cupidité individuelle et intérêts de la coopération collective. Pour cette raison, il est devenu un des modèles les plus utilisés en sociologie, biologie et économie.

### 1.2 Le dilemme du prisonnier itéré

Le dilemme du prisonnier devient plus intéressant et plus réaliste lorsque la durée de l'interaction n'est pas connue. On peut alors envisager de se souvenir du comportement d'un joueur à son égard et développer une stratégie en rapport. Par exemple, si je sais que mon adversaire ne coopère jamais, mon intérêt sera de ne pas coopérer non plus, sous peine d'être systématiquement floué. Par contre si je sais que mon adversaire coopérera toujours quoi qu'il arrive, j'aurai intérêt à être vicieux et ne jamais coopérer pour maximiser mon gain.

### 1.3 L'interaction de tribus

Considérons deux tribus d'indigènes partant à la chasse. Ces deux tribus peuvent choisir entre coopérer ou trahir la tribu adverse lors d'une confrontation. Lorsque la situation du dilemme est itérée,

le jeu devient très intéressant, car la question ne se pose plus sous la forme : *trahir ou coopérer ?*, mais sous la forme : *quelle stratégie faut-il adopter en fonction du comportement passé de l'entité adverse ?*.

Nous supposons que les deux tribus ne peuvent pas passer d'accord. La seule information qu'une tribu connaît sur l'autre est son comportement passé lors des coups précédents. Les décisions des deux tribus lors de la partie sont prises simultanément. Le nombre de parties n'est pas connu à l'avance.

Décrivons la variante adoptée de manière plus abstraite : deux entités peuvent choisir entre coopérer (notation  $c$ ) ou trahir (notation  $t$ ). Si l'une trahit et l'autre coopère (partie  $[t, c]$ ), celle qui trahit obtient un gain de  $T$  unités et celle qui coopère (et s'est donc fait duper) obtient un gain de  $D$  unités. Lorsque les deux entités coopèrent (partie  $[c, c]$ ), elles gagnent chacune  $C$  unités en récompense de leur association. Quand elles trahissent toutes les deux (partie  $[t, t]$ ), elles gagnent  $P$  unités pour s'être laissées piéger mutuellement. Le choix des coefficients  $T, D, C$  et  $P$  n'est pas fortuit. Conformément aux n°181 de *POUR LA SCIENCE* nous prenons :  $T = 5, D = 0, C = 3, P = 1$ .

## 2 Description du projet

### 2.1 L'application à réaliser

L'application à réaliser doit permettre de mettre en œuvre des rencontres interactives entre deux joueurs se basant sur le dilemme du prisonnier itéré selon les modalités décrites en 1.3.

L'application est implémentée en Java en mode client-serveur.

Une rencontre se joue en  $n$  tours,  $n$  étant défini par l'utilisateur qui a initié la rencontre.

Le deuxième joueur se joint à la partie.

A chaque tour, chaque joueur donne sa décision, son score est calculé ainsi que celui de l'adversaire, les deux sont affichés.

Le score réalisé par un joueur à la fin de la rencontre est la somme de ses points récoltés lors de chacune des confrontations.

À tout moment un joueur peut choisir d'abandonner la rencontre, il choisit alors une stratégie automatique parmi les stratégies décrites en 2.2.

L'autre joueur continue à jouer normalement en mode interactif contre le serveur.

### 2.2 Exemples de stratégie

1. **Donnant donnant** - Jouer comme le dernier coup de l'adversaire
2. **Donnant donnant / aléatoire** - Jouer comme le dernier coup de l'adversaire, mais jouer parfois un coup au hasard.\*
3. **Donnant pour deux donnants et aléatoire** - Comme donnant donnant sauf que l'adversaire doit faire le même choix deux fois de suite avant la réciprocité. Joue parfois un coup au hasard.\*
4. **Donnant pour deux donnants** - Comme donnant donnant sauf que l'adversaire doit faire le même choix deux fois de suite avant la réciprocité.
5. **Sondeur naïf** - Jouer comme le dernier coup de l'adversaire, mais parfois trahir au lieu de coopérer.\*
6. **Sondeur repentant** - Jouer comme le dernier coup de l'adversaire, mais parfois trahir au lieu de coopérer. Si l'adversaire trahit en réponse au test, se montrer repentant en coopérant immédiatement.\*
7. **Pacificateur naïf** - Jouer comme le dernier coup de l'adversaire, mais faire parfois la paix en coopérant au lieu de trahir.\*
8. **Vrai pacificateur** - Coopérer si l'adversaire ne trahit pas deux fois de suite. Trahir alors immédiatement, mais essayer parfois de faire la paix en coopérant au lieu de trahir.\*
9. **Aléatoire** - Trahir ou coopérer avec une probabilité de 50%
10. **Toujours trahir**
11. **Toujours coopérer**
12. **Rancunier** - Coopérer jusqu'à ce que l'adversaire trahisse. Ensuite toujours trahir.
13. **Pavlov** - Si 5 ou 3 points ont été obtenus au tour précédent, répéter le dernier choix.
14. **Pavlov / Aléatoire** - Si 5 ou 3 points ont été obtenus au tour précédent, répéter le dernier choix, mais faire parfois des choix aléatoires.\*
15. **Adaptatif** - Commencer avec  $c, c, c, c, c, t, t, t, t, t$ . Choisir ensuite le choix ( $t$  ou  $c$ ) qui a donné le meilleur score en moyenne. Recalculer ce choix après chaque coup.

16. **Graduel** - Coopérer jusqu'à ce que l'adversaire trahisse; dans ce cas, trahir autant de fois que l'adversaire a déjà trahi dans la partie, puis continuer par deux coopérations.
17. **Donnant donnant soupçonneux** - Comme donnant donnant, mais commencer par trahir
18. **Rancunier doux** - Coopérer jusqu'à ce que l'adversaire trahisse; dans ce cas l'adversaire est puni par t,t,t,t,c,c.

\* *Stratégies avec intervention du hasard*