

# TP1 Reinforcement Learning

December 7, 2016

## Objectif

L'objectif de cette séance de TP est d'implémenter l'algorithme de Q-Learning dans un jeu de *Tic tac toe*, d'observer l'amélioration de la stratégie d'un agent au cours de son apprentissage et de faire varier les paramètres pour améliorer les performances.

## Questions préliminaires

Modélisez le jeu du *Tic tac toe* sous la forme d'un problème d'apprentissage : quels sont les états, les actions, les récompenses ?

Combien d'états dénombrez vous ?

## Prise en main

Téléchargez l'archive <https://cointe.users.greyc.fr/download/tictactoe.zip>

Créez un projet Java avec eclipse et placez les fichiers sources dans le dossier src. Lancez le programme pour vérifier que cela compile et effectuez rapidement une partie à deux joueurs dans la console. Comme le montre la figure 1, chaque case est identifiée par un nombre pour permettre au joueur de la désigner.

```
-----  
| 1 | 2 | 3 |  
-----  
| 4 | 5 | 6 |  
-----  
| 7 | 8 | 9 |  
-----
```

Agent 1, à vous de jouer. Quelle case voulez-vous ?

Figure 1: Un tic tac toe rudimentaire

Le programme se présente de la manière suivante :

- Une classe `Grid` représente la grille sous forme d'une (une matrice  $3 \times 3$ ) et propose quelques méthodes :
  - `getCell(x)` renvoie le contenu de la cellule numéro `x`
  - `getAllCells()` renvoie une copie de la grille sous forme d'une liste de listes
  - `isEmptyCell(x)` renvoie `true` si la cellule numéro `x` n'est pas encore prise par un joueur, `false` sinon
  - `isWon()` renvoie `true` si l'un des deux joueurs a remporté la partie, `false` sinon
  - `put(x,s)` insère le symbole `s` dans la cellule numéro `x`

- Une classe abstraite **Agent** dotée d'une seule méthode abstraite :

- `play()` appelée lorsque l'agent est à son tour de jeu.

- Une classe **HumanAgent**, héritant de **Agent** et implémentant la méthode `play` pour permettre à un utilisateur de jouer dans la console.

Que se passe-t-il dans l'état actuel du programme si les joueurs font match nul ?

Commencez par implémenter la méthode `public boolean isFull()` indiquant si le jeu est terminé et modifiez le programme principal pour mettre fin à la partie.

## Q-Learning

Implémentez maintenant une nouvelle classe **QLearningAgent** héritant de la classe abstraite **Agent** jouant selon l'algorithme d'apprentissage Q-Learning. Ne modifiez pas les autres classes sans en discuter au préalable avec les autres élèves.

Une fois votre classe implémentée, observez son comportement sur une série de parties. Qu'observez-vous ? Vous pouvez faire s'affronter vos programmes pour comparer vos implémentations et leur paramétrage.