

Projets de développement informatique : Labo 5

Agent logiciel et intelligence artificielle : Tic-Tac-Toe et Puissance 4

Durant ce labo, vous allez découvrir la notion d'intelligence artificielle pour représenter un joueur d'un jeu de plateau. À l'aide d'une librairie que l'on vous fournit, vous allez devoir représenter des agents logiciels qui définissent le comportement d'un joueur « *intelligent* ». Ce labo est l'occasion de prendre en main cette librairie et de développer un joueur intelligent pour les jeux Tic-Tac-Toe et puissance 4.

1 Librairie d'IA

Examinons maintenant la librairie que nous allons utiliser pour coder des intelligences artificielles. Vous la trouverez sur le dépôt Git du cours, dans le dossier `AIproject`¹. En plus de la librairie, qui se trouve dans le fichier `lib/game.py`, vous trouverez deux exemples de jeu : *Tic-tac-toe* et *Puissance 4*, se trouvant respectivement dans les fichiers `tictactoe.py` et `connectfour.py`.

Voyons d'abord comment lancer le premier exemple. Pour cela, il vous faut d'abord démarrer le serveur du jeu avec la commande suivante² :

```
$ ./tictactoe.py server --verbose
Game server listening on MacBook-Pro-de-Sebastien-3.local:5000
```

Il faut ensuite lancer les deux joueurs. On commence par lancer le premier joueur avec la commande suivante, qui doit spécifier le nom du joueur :

```
$ ./tictactoe.py client --verbose Théo
Connected to the game server on 127.0.0.1:5000
```

À ce moment, on peut voir dans la fenêtre du serveur qu'un joueur s'est connecté. Il ne nous reste plus qu'à lancer le second joueur, après quoi la partie démarrera immédiatement :

```
$ ./tictactoe.py client --verbose Sébastien
Connected to the game server on 127.0.0.1:5000
Game started
Player's turn to play
State of the game: 0 None None None None None None None None
Next move: 1
Player's turn to play
State of the game: 0 1 0 None None None None None None
Next move: 3
Player's turn to play
State of the game: 0 1 0 1 0 None None None None
Next move: 5
You lost the game
```

Examinez le fichier `tictactoe.py` pour comprendre comment le client a été codé. Le client est représenté par la classe `TicTacToeClient` dans laquelle deux méthodes doivent se retrouver : `_handle` et `_nextmove`.

1. Il s'agit d'une librairie en cours de développement, toute suggestion d'amélioration ou correction d'éventuels bugs est la bienvenue et souhaitable.

2. L'option `verbose` permet de rendre le serveur bavard, ce qui fait qu'il affichera plus d'informations dans votre console.

Vous ne devriez pas avoir besoin de la première, mais devez vous concentrer sur la seconde. C'est en effet dans cette dernière que vous devez déterminer le prochain coup que vous désirez faire en fonction de l'état reçu en paramètre.

2 Tic-Tac-Toe et puissance 4

Afin de prendre en main la librairie, écrivez une intelligence artificielle plus maligne que celle proposée en exemple pour un joueur de Tic-tac-toe et faites ensuite de même pour le puissance 4.