

IA et Jeux – TP N°1 : Ms. PacMan



Objectif du TP

L'objectif de ce TP est de découvrir le framework MsPacMan que nous utiliserons pendant tout le module.

LA PLATEFORME MS. PACMAN

[Ms. PacMan](#) est une plateforme qui simule le jeu [PacMan](#) et qui permet de créer des bots intelligents. Ce jeu a été très utilisé pour mettre au point un grand nombre d'IA.

REDIGER UN COMPTE-RENDU

1. A la fin de ce TP, vous devrez rédiger un compte-rendu. Pour savoir ce qu'il doit/devrait contenir, lisez *attentivement* ce texte <https://guillaume.piolle.fr/doc/tp.pdf>.

INSTALLATION DE LA PLATEFORME

2. Téléchargez le fichier [zip](#) de la plateforme et décompressez le.
3. Ouvrez Eclipse, créez un nouveau projet Java en référençant le répertoire `mspacman-master` décompressé.
4. Téléchargez la librairie [commons-lang](#) de Apache Commons et ajoutez la librairie à votre projet Eclipse. Elle nous sera utile pour travailler sur les tableaux avec la classe [ArrayUtils](#).
5. Téléchargez la librairie [commons-math](#) de Apache et ajoutez la librairie à votre projet Eclipse. Elle pourra nous servir pour faire des calculs mathématiques (tests statistiques avec la classe `TestUtils` ...).



DECOUVRIR L'ENVIRONNEMENT

6. Exécutez la classe `Executor`. Cela va peut-être un peu vite pour vous ? Modifiez la valeur de la variable `delay`. Pensez à tester avec une valeur nulle : c'est ce qu'on appelle un jeu rapide !
7. Exécutez plusieurs fois le jeu et observez le parcours de PacMan. Qu'en déduisez-vous ?
8. Selon vous, à quoi correspond S, L et T en bas de l'écran ?
9. Modifier le code source en changeant la classe qui contrôle PacMan en `RandomPacMan`. Que constatez-vous ? Modifiez ensuite avec `RandomNonRevPacMan`. Essayez de comprendre la différence entre les deux stratégies en analysant le code source. Expliquez !
10. Allez, vous en mourrez d'envie : lancer un jeu auquel vous pouvez contrôler Ms. PacMan au clavier ! Mais pas trop longtemps, svp car il vous reste encore beaucoup de travail.

11. Lancer la première expérimentation intitulée « Starter Pacman vs Legacy2Thereconing ». A quoi correspondent ces chiffres ?
12. Exécuter également en même temps les deux expérimentations suivantes mais en changeant `RandomPacMan` en `RandomNonRevPacMan`. N'hésitez pas à modifier la variable `numTrials`. Que pouvez-vous en déduire ?
13. Créez une nouvelle méthode nommée `runExperiment2` qui contrairement à `runExperiment` retourne les valeurs numériques de chaque exécution dans un tableau (de réels) plutôt que de les afficher.
14. Essayez de vérifier vos intuitions de la question 12 en appliquant un test statistique, bien plus scientifique qu'une simple intuition visuelle, avec la méthode `TestUtils.pairedTTest`. Cela correspond au test de Student ou t-test avec données appariées par rapport à la graine du générateur du nombre aléatoire. Il est également possible de choisir la valeur de α (alpha). Ce test est expliqué ici [ici](#) ou [là](#). Le résultat précise si les deux moyennes sont significativement différentes ou non, ou bien renvoie la valeur de la p-value en fonction de la méthode que vous utilisez.
15. Jouer une partie en enregistrant la partie dans un fichier. Pour cela, identifiez le code dans la classe `Executor` qui permet de le faire et exécuter le. Analyser le fichier texte généré. Est-ce que vous y comprenez quelque chose ? Faites rejouer la partie que vous venez de jouer par `Executor`. Modifiez les méthodes qui utilisent ou génèrent ce type de fichier en précisant le nom du paramètre et sa valeur correspondante. Exemple de ligne qui devra être générée ou lue : `mazeIndex=0, totalTime=6, score=0, currentLevelTime=6, levelCount=0, pacman.currentNodeIndex=974, pacman.lastMoveMade=LEFT, pacman.numberOfLivesRemaining=3, pacman.hasReceivedExtraLife=false, timeOfLastGlobalReversal=-1, pacmanWasEaten=false...` Pour cela, utilisez une structure de données de type `HashMap<Key=nom du paramètre, Value=valeur du paramètre>`.