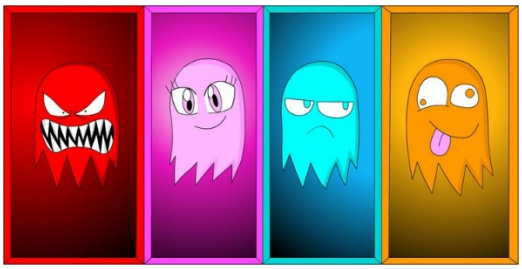
IA et Jeux – TP N°1 : Ms. PacMan



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.

Rédiger un Compte-Rendu

1. À 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` …).

Découvrir 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 ». À 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.

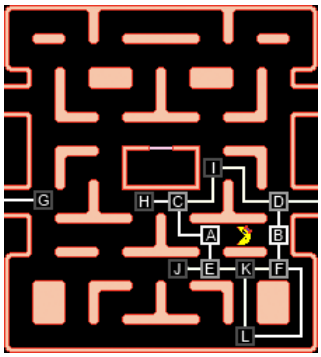
14. Jouer une partie en enregistrant la partie dans un fichier. Pour cela, identifiez le code dans la classe `Executor` qui permet de le faire et exécuter le. Analyser le fichier texte généré. Est-ce que vous y comprenez quelque chose ? Faites rejouer la partie que vous venez de jouer par `Executor` en chargeant le texte de la partie jouée. Modifiez les méthodes qui utilisent ou génèrent ce type de fichier en ajoutant dans le fichier texte généré 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>`.

DÉBOGUAGE GRAPHIQUE

16. Lancer une exécution visuelle avec `NearestPillPacMan` puis ensuite avec `NearestPillPacManVS`. Décommentez les lignes qui ajoutent des lignes vers les fantômes et exécutez. Faites une copie de cette dernière classe en la renommant `NearestPillOrGhostsPacManVS` et placez la dans le package `entries.pacman`. La stratégie de Ms. Pacman consiste à aller vers les pacGommes les plus proches sans se soucier des fantômes. Modifiez cette classe pour que PacMan se focalise sur les fantômes s'ils sont comestibles ou vers les pacgommes sinon.
17. Ajouter en magenta, le plus court chemin calculé avec la technique A* implémentée en utilisant la métrique euclidienne.
18. Supprimer les affichages visuels que vous avez pu ajouter. Utilisez la méthode `getJunctionIndices` pour afficher les nœuds de jonction pour voir à quoi ils correspondent. Sachez que la méthode `isJunction(nodeIndex)` permet de savoir si le nœud `nodeIndex` en est ou pas.

UN PEU DE RÉCURSIVITÉ !

19. *The last but not the least* ! Ecrivez la méthode `getClosestJunctionIndices` qui permet de connaitre les nœuds de jonction vers lesquels Ms. PacMan peut aller. Attention, il ne suffit pas de prendre les 2 ou 3 nœuds de jonction les plus proches de Pacman. La preuve sur la figure ci-contre : Les seuls nœuds possibles sont A et B alors qu'en utilisant les distances, cela aurait pu être A et K ou B et K. Par ailleurs, imaginez que Pacman se trouve en E, alors le résultat correct attendu est {A, J, K}. *Indice* : Il existe la méthode `getPossibleMoves` à partir de la position courante. De plus, vous pourriez avoir besoin de connaitre la position précédente de PacMan et lors du premier appel à la méthode, préciser que la position précédente est égale à la position courante…



Sources pour ce TP :

- gameaibook.org
- Why AI Researchers Love Playing Pac-Man?