

# 8INF846 – Intelligence Artificielle

## Travail No. 3

### Agent logique

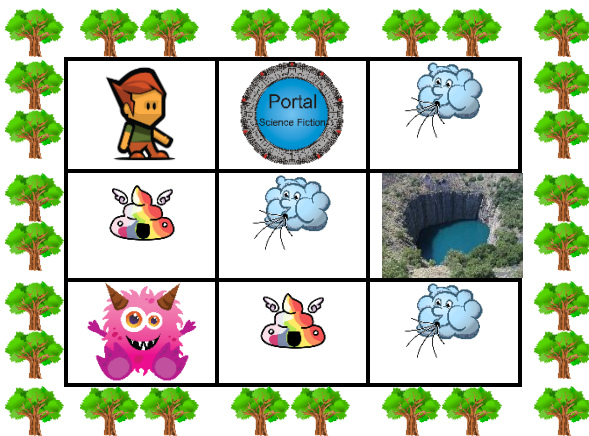
---

#### *Mise en situation:*

Depuis l'incident du Sudoku, votre conjoint(e) ne vous contredit plus lors de discussions. Enfin, vous vous demandez si il/elle boude, mais ça fait un peu votre affaire puisque vous pouvez écouter « Tout le monde en parle » en paix le dimanche soir et même mettre CNN pour suivre la politique américaine (vous avez appris plusieurs choses pertinentes : Clinton est une féministe qui porte le nom de son mari infidèle et Trump aime tellement son pays qu'il veut ériger des murs autour pour le protéger). Malheureusement, votre conjoint(e) gardait en fait de la rancœur face aux événements des dernières semaines (après tout, vous avez perdu ses bijoux avec un robot mal programmé et insulté son intelligence en trichant au Sudoku). Quoi qu'il en soit, le 21 mars, alors que vous vous rendiez dans une fête pour célébrer le printemps (toutes les occasions sont bonnes pour célébrer), il/elle a arrêté la voiture, vous a assommé et vous vous êtes réveillé au milieu de la forêt magique la plus dangereuse au monde. Heureusement, vous aviez votre cellulaire et vous décidez donc de programmer un système intelligent pour vous aider à trouver la sortie.

#### *Détails du travail à effectuer :*

Vous devez implémenter un agent intelligent utilisant la logique afin de naviguer dans la forêt et de trouver la sortie. La forêt doit être générée aléatoirement. La forêt initiale contient 3x3 cases, la seconde 4x4 et ainsi de suite. Voici un exemple de carte générée :



En principe, il ne devrait y avoir qu'un bouton sur l'application, soit « Bouger ». Le bouton lance un cycle d'inférence permettant de déterminer la case où se déplacer. Vous ne voyez qu'une case à la fois. L'inférence doit se faire grâce à une modélisation dans un langage logique. Évidemment, vous pouvez faire celle-ci à l'aide d'un langage de programmation.

#### Mesure de performance :

- Chaque sortie trouvée :  $+10 \times \text{nombre de cases}$
- Mort :  $-10 \times \text{nombre de cases}$
- Mouvement :  $-1$
- Utilisation de roches :  $-10$

#### Environnement :

- Les cases adjacentes à un monstre sentent mauvais
- Les cases adjacentes à une crevasse sont venteuses

- Un portail magique brille si la case contient la sortie
- Tirer une roche tue un monstre dans la case directement devant vous
- Tirer utilise exactement une roche
- Sortir permet de prendre un portail (si présent) et de se téléporter vers une nouvelle carte

Effecteurs : Gauche, Droite, Avant, Arrière, Tirer, Sortir

Capteurs : Odeur, Vent, Lumière

**Type d'environnement :**

Partiellement observable (on ne voit que notre case), déterministe, séquentielle (chaque épisode est indépendant), statique (la carte ne change pas), discret (temps et espace finis), simple agent (les monstres sont passifs).

**Contraintes à respecter :**

1. Le jeu est infini.
2. L'intelligence est implémentée grâce à un formalisme logique (10% de bonus si vous utilisez Prolog, même si c'est avec un interpréteur dans un autre langage comme Java ou Python).
3. Vous devez utiliser la modélisation agent (tel que vu en début de trimestre).
4. Vous devez intégrer des notions probabilistes.

---

Vous devez me remettre une copie électronique de votre code incluant une page titre avec votre nom, votre code permanent et le titre du cours. Si vous avez des indications particulières concernant l'exécution, la compilation ou autres, veuillez ajouter une page à cet effet juste après la page titre. Pour la remise, veuillez respecter la procédure expliquée dans le document « Procédure de remise.pdf ».

Date limite de remise : **4 avril 2017 à 23h59**

Travail en équipe de 2 ou 3.

**Bon travail !**