

8INF846 – Intelligence Artificielle

Travail No. 1

Création d'un agent aspirateur

Travail à effectuer (sommaire):

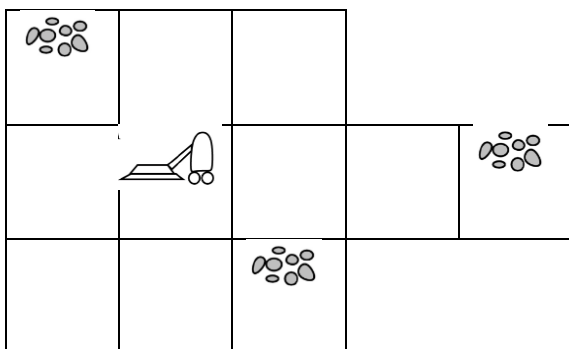
Vous possédez un grand manoir constitué de nombreuses pièces qui ne cessent de constamment se salir. Depuis que vos employés d'entretien se sont syndiqués, ils demandent à travailler des heures épouvantables (de 8 à 4 vous imaginez!) et d'obtenir des pauses durant la journée. Tant pis pour eux! Vous décidez de commander un Aspirobot T-0.1 enfin de remplacer tous ces paresseux. Après tout, la compagnie vous a garanti que celui-ci éviterait d'aspirer vos bijoux dispendieux (ce serait triste de perdre un diamant à 15000\$ pour un peu de poussière) et que de façon automatique il maintiendrait un maximum de pièces propres en tout temps sans gaspiller d'électricité!

Détails du travail à effectuer :

Environnement :

L'environnement dans lequel évolue l'agent Aspirobot T-0.1 sera constitué d'une carte contenant 11 pièces disposées telles qu'illustrées ci-dessous. Tel que défini au cours, l'environnement devrait contenir tous les éléments passifs de votre programme (carte, poussière, bijoux, etc.). L'environnement devrait aussi contenir minimalement une mesure de performance que l'agent peut consulter (via ses capteurs). L'environnement devrait être une boucle infinie (ou encore un événement programmé pour s'exécuter sporadiquement) qui aléatoirement génère soit de la poussière dans une case aléatoire ou encore un bijou. C'est à vous de décider des probabilités. Une case peut contenir à la fois de la poussière et un bijou.

```
While (gameIsRunning()){  
    If (shouldThereBeANewDirtySpace())  
        GenerateDirt()  
    If (shouldThereBeANewLostJewel())  
        GenerateJewel()  
}
```



Agent :

L'agent Aspirobot T-0.1 possède un certain nombre de fonctionnalités. Il peut aspirer (ce qui a pour effet d'aspirer à la fois la poussière et les bijoux), ramasser un élément (pour ramasser un bijou idéalement!) et se déplacer (haut, bas, gauche et droite). Chacune de ses actions lui coûte 1 unité d'électricité (attention, rappelez-vous que vous êtes un propriétaire de manoir ~~cheap~~ heum... disons qui n'aime pas gaspiller de l'argent). Le robot ne peut pas voir plus loin que la

pièce courante dans laquelle il se situe. Il peut cependant avec une mémoire de ce qu'il a perçu (y compris la configuration de la carte). Le robot doit être implémenté sous les principes vus en classe. Il doit donc observer l'environnement avec ses capteurs et agir sur celui-ci avec des effecteurs (je vous conseille carrément de faire des classes capteurs et effecteurs). Votre robot devrait être de type « basé sur les buts ». Même si le problème est très simple, il est important que celui-ci contienne les éléments de base (son état interne devrait contenir un état mental sous la forme BDI « Beliefs-Desires-Intentions »). Il devrait lui aussi être implémenté comme une boucle infinie respectant l'idée de base vue en cours (voir pseudo-code simple ci-dessous).

```
While (amIAlive()){  
  ObserveEnvironmentWithAllMySensors()  
  UpdateMyState()  
  ChooseAnAction()  
  justDoIt()  
}
```

Contraintes à respecter :

1. L'agent et l'environnement doivent s'exécuter sur deux fils d'exécution différents.
2. De la poussière et des bijoux doivent être générés sporadiquement par l'environnement.
3. L'agent dépense une unité d'électricité par action.
4. Si l'agent aspire et qu'un bijou se trouve à cet endroit, l'environnement devrait considérer que celui-ci a été aspiré (le robot n'a pas nécessairement à le savoir, puisqu'il ne l'avait pas perçu).
5. L'agent doit posséder un état mental BDI (attention, la partie I sera probablement presque vide).

L'implémentation devra être cohérente avec les notions vues en classe. La modélisation des agents, de l'environnement, des actions, etc., devra s'appuyer sur les modèles vus en classe.

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. Votre travail devra être accompagné d'un rapport expliquant votre modélisation et comment chacun des éléments est conforme à l'architecture présentée dans les notes de cours. Ce rapport devra positionner clairement votre implémentation : décrire le type d'environnement (propriétés), le type d'agent, la fonction d'agent, modélisation de l'action, de la perception, etc.

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 : **6 Octobre 2016 à 23h59**

Bon travail !