Projet Aspirateur v2

Terral Naomie
Rodriguez Charlotte
Geshkovski Borjan

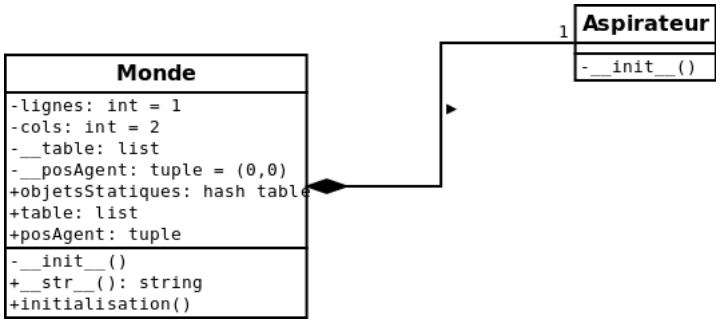
06.02.2016

Abstract

Ceci est la deuxième version de notre projet Aspirator stochy debilum. Le code est organisé de manière modulaire.

Organisation des données

Pour lancer le programme il fauat exécuter le fichier launch.py. Le module fonctionnel (monde.py) se trouvent dans le dossier data.



- `objetsStatiques`: dictionnaire dont la clé est l'identifiant de l'élément contenu dans le monde. La clé fait référence à un tuple contenant la signification de l'élément et le caractère qu'on lui associe;
- `Monde()` : contient notre environnement. Nous allons décrire les différents attributs et méthodes:
  - `__init__(aspirateur,lignes,colonnes)` : Constructeur de l'envrironnement;
  - `objetsStatiques` : Renvoie objetsStatiques localement dans le monde;
  - `table` : Renvoie l'état du monde;
  - `posAgent` : Renvoie la position de l'aspirateur;
  - `agent` : Renvoie l'agent (un aspirateur);
  - `historique` : Renvoie l'ensemble d'etats; durant une simulation;
  - `perfGlobale` : Renvoie la performance globale de l'aspirateur;

  - `applyChoix(action)` : Change le monde en fonction d'une action choisie par l'aspirateur, et renvoie une evaluation de cette action;
  - `getPerception(capteurs)` : Renvoie le contenu dans les capteurs de l'aspirateur;
  - `__str__` : Affichage du monde (état et aspirateur);
  - `step` : Une etape pendant une simulation;
  - `simulation(n)` : Simulateur d'un environnement de n etapes;
  - `initialisation` : Initialise la position de l'aspirateur et l'état du monde de façon stochastique.
  - `updateWorld`: MaJ du monde.

- `Aspirateur()`
  - `__init__(capteurs, actions)` : Constructeur d'aspirateur ;
  - `vivant` : Dead or alive.
  - `capteurs` : Les cases que l'aspirateur peut voir ;
  - `actions` : Les actions que l'aspirateur peut executer ;
  - `getDecision(content)` : L'aspirateur choisit quoi faire en fonction du contenu de la case ;
  - `getEvaluation()` : Feedback ;
  - `setReward(reward)` : Reward mechanism ;

- `AspiClairVoyant()` :

- `AspiVoyant()`