CS 335 Introduction to Artificial Intelligence

Fall 2016

Northeastern Illinois University

Assignment #2: Creating A Vacuum Cleaner Agent

Due Tuesday September 20, 2016 by midnight

Instructor: Oguzcan Adabuk

The goal of this assignment is to create a vacuum cleaning agent that will move in a predefined environment until either all the dirt is cleaned or the agent runs out of battery power.

Agent's performance is calculated by the points it gets by cleaning dirty tiles minus the battery power it uses. Agent receives 15 points per dirty tiles it cleans and uses 1 battery power per move. For example if the agent has cleaned two tiles and had to move 32 times than the performance would be : performance = 15 X 2 – 32 = -2.

Agent has a battery capacity of 50, so if the agent cannot clean the tiles in 50 moves then it should run out of battery, stop and display an error message such as *"The agent ran out of battery before cleaning all the tiles...".*

This assignment consists of three Java classes and an agent interface. The Environment and the Test classes are provided to you along with the Agent Interface.

The Environment class creates an environment for the agent to run on. It creates a 2D array that represents the agent's environment. Paths that the agent can take are marked as "P", areas with obstacles are marked as "O" and the dirty places are marked as "D". CreateEnvironment() method randomly creates a new environment.

The Test class creates a new environment and a new agent. Agent's constructor should setup the agent and start it running.

The agent interface gives you the methods you need to implement within the agent class.

DO NOT make any changes to the Environment and the Test classes and the agent interface!!!

You only need to implement the agent class.

The Agent class needs to implement every method inside the agent interface.

The Agent class has following required instance variables:

- *int x*: Agent's row number.

- *int y*: Agent's column number.
- *int battery*: This represents the battery power agent has used. It should be incremented every time the agent moves.
- *int performance*: This variable represents the agents performance. Measured by (number of tiles cleaned X 15 – battery).
- *Environment environment:* This variable contains the environment the agent runs in. It is passed into the agent via its constructor when an instance of the agent is created. This way the agent can access the 2D array world that represents the environment.

The Agent constructor is called inside the Test class. It takes an x and a y values for the Agent's initial coordinates and an instance of an environment object.

How to Run The Agent?

-You must implement the methods that are inside the AgentInterface:

```
public interface AgentInterface {
        public void MoveLeft();
        public void MoveRight();
        public void MoveDown();
        public void MoveUp();
        public void MoveLeftUp();
        public void MoveLeftDown();
        public void MoveRightUp();
        public void MoveRightDown();
        public void GetNewDirection();
        public boolean CheckForDirt(String [][] world);
        public void CleanDirt(String [][] world);
        public void Run(String [][] world);
        public void PrintWorld(String [][] world);
}
```

-The agent must run in a loop. The loop continues until either all the tiles are clean or the agent has used 50 units of battery power.

-The agent must choose a random direction and continue to move in that direction until either it hits an obstacle or the boundaries of the environment. Then the agent should get another movable direction and continue moving.

-Every time the agent moves to a new tile, it should mark its current position with an "A" and then check whether the tile is clean or not. If the tile is dirty, the agent should clean it. When the agent cleans a tile, it should increase its performance by 15 and print out a message that displays which tile is cleaned. Take the necessary actions to prevent the agent from cleaning the same tiles more than once.

-After every move the agent should display the current state of its environment, its position, performance and battery usage in the following format:

```
A P P P P
A A P O D
A P A P P
A D P P P
A P P O P
Agent at: [3,3]. Battery power used: 19 Performance: -19
```

-The agent should display an informative message when it stops running. The message should let the user know that the agent stop running either because it ran out of battery or cleaned up all the tiles. Message format should be like this:

```
The agent ran out of battery before cleaning all the tiles...
Battery power used: 50
Performance: -35
```

OR

```
All tiles are clean!
Battery power used: 30
Performance: 0
```

How to Submit Your Assignment?

Compress (ZIP) the folder that contains your classes and submit it to the dropbox folder for this assignment. You should submit only 1 folder that has the same number of classes I provided you. I will deduct points from assignments that do not follow these instructions.