CSCI 312 -- Introduction to Artificial Intelligence

Project -- Vacuum Cleaner Agent

Name:	Hayden Holbrook	Comparison Partner(s)	Tom Philipps
Completion Date:	1-8 Sep 17, 9-10 Sep 24		
Assigned Percept Type (Bump or Location)	Bump		

General Description:

Using performance-measuring environment simulator for the vacuum cleaner world, create and evaluate three agent models: a randomize reflex agent, a reflex-model agent using bump sensors, and a reflex-model agent using a location indicator. This project modifies and incorporates exercises 2.11 through 2.13 on page 63 of the Russell and Norvig text.

Deadlines:

1. Complete Tasks #1 through #8 by 11:55pm Monday September 18, 2023 **ALL LATE SUBMISSIONS WILL RECEIVE A GRADE OF ZERO!!!**
2. Submit completed project by 11:55pm Monday September 25, 2023 **ALL LATE SUBMISSIONS WILL RECEIVE A GRADE OF ZERO!!!**

Vacuum Cleaner World Specifications:

1. The geography of the environment (its extend, boundaries, obstacles) is unknown to agent. The agent does know the geometry is a grid with top left coordinates of (0,0).
2. Agent is circular in shape.
3. Agent does not know its initial location.
4. The initial dirt distribution is unknown to the agent.
5. The agent may perform the following actions:
   a.) VacuumAction.LEFT — Agent moves 1 cell horizontally in the negative x direction.
   b.) VacuumAction.RIGHT — Agent moves 1 cell horizontally in the positive x direction.
   c.) VacuumAction.FORWARD — Agent moves 1 cell vertically n the negative y direction (towards the top of grid).
   d.) VacuumAction.BACK — Agent moves 1 cell vertically n the positive y direction (towards the bottom of grid).
   e.) VacuumAction.SUCK — Removes dirt from cell located at agent's current position.
   f.) VacuumAction.STOP — Agent ceases all actions.

6. Performance :

   Given the following at time step *t*:

dirtSquares(t)	The number of dirty squares at time t.
initialDirtySquares	The number of dirty squares at start of simulation.
initialSquares	The number of open squares in the environment.
targetTime	The hypothesized optimal time for simulation. For this project, targetTime = 2*initialSquares + 1
maxTimeSteps	The number of iterations for simulation

The agent performance measure in t-steps, Performance(t), is:

$$Performance(t) = 0.75 * cleanPercent(t) + 0.25 * timePercent(t) * cleanPercent(t)$$

where $cleanPercent(t) = 1.0 - \frac{dirtSquares(t)}{initialDirtySquares}$

$$timePercent(t) = \frac{maxTimeSteps - t + targetTime}{maxTimeSteps}$$

7. Percept Information:
   a.) Bump Percept

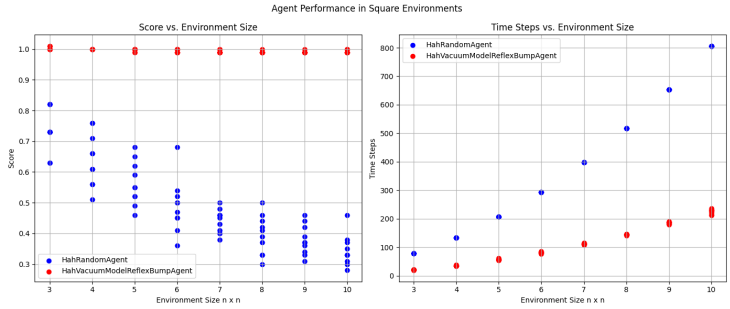
i.)	clock()	Returns the current time in simulation.
ii.)	dirtSensor()	Returns a status of CLEAN or DIRTY.
iii.)	bumpSensor(action)	For a given action, bumpSensor returns TRUE if action will cause agent to bump into an obstacle. Returns FALSE otherwise.

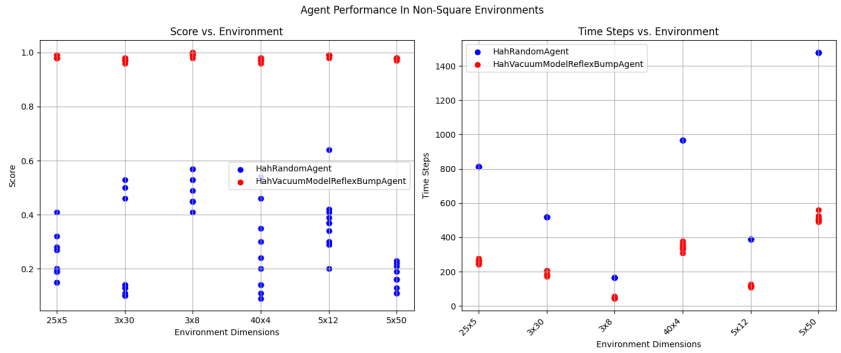
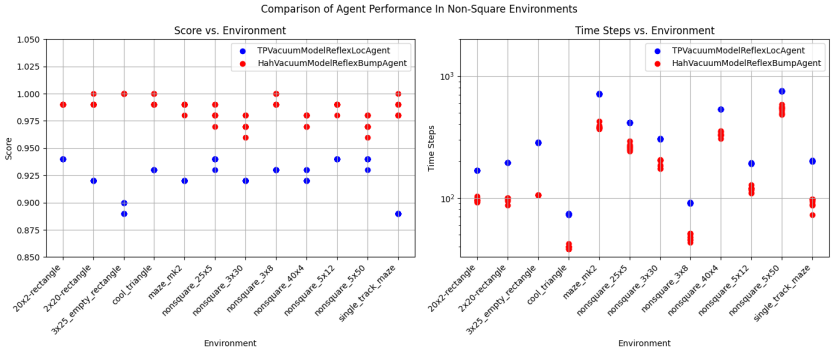
   b.) Location Percept

i.)	clock()	Returns the current time in simulation.
ii.)	dirtSensor()	Returns a status of CLEAN or DIRTY.
iii.)	getRow()	Returns the row of the agent's current location.
iv.)	getCol()	Returns the col of the agent's current location.

Task List:

	Task	**Possible Points**	**Scoring Notes**
1.	Download VacuumProjectStarter2023.zip from Moodle Review the documentation and the VacuumAgent Forum for instructions on how to run/modify the simulation. Command line options for VacuumAgentDriver: -h : display help info. All other options ignored. -d w h : Set environment's maxWidth to integer w and maxHeight to integer h. -a x y : Set agent's initial floor position to first location starting at x,y in row-major order. -p prob : Set probability of a clean square becoming dirty to double prob. -P m : Set target performance to double m. -g filename : Create environment's floor using the pattern given in FILE filename. Has higher precedence		

	than option -d. -t n : Set simulation's maxTime steps to n. -B : Use BUMP percept for agent simulation -L : Use LOCATION percept for agent simulation -v : Use visualizer -A className : Implement agent using className.class		
2.	Create a randomized reflex agent class that extends the class VacuumAgent. Name this class XyzRandomAgent where "Xyz" is prefix containing your initials. Your getAction method should use your assigned percept.	**10**	To receive credit, agent must execute
3.	Test your XyzRandomAgent on environments on 10 n x n square environments start at n=3. Use the -t option to adjust the maxTimeSteps that will allow your agent to complete clean the floor. Graph the performance results. Place your chart in this cell. 	**10**	
4.	Design an environment in which your XyzRandomAgent performs poorly.	**5**	Agent should perform significantly worse than in the above environments.
5.	Create an XyzReflexModelAgent that extends the class Vacuum Agent. Name this class XyzReflexModelAgent where "Xyz" is prefix containing your initials. Your getAction method should use your assigned percept.	**15**	To receive credit this agent must receive a minimum performance score of 64% .
6.	Test your XyzReflexModelAgent on the same environments from step 3. Compare the performances of XyzRandomAgent and XyzReflectModelAgent. This time do not use the -t option to adjust the maxTimeSteps. You will post your chart to a forum in step #8. 	**5**	

7.	Test your XyzReflextModelAgent on five non-square environments. Show your results in a chart. 		
8.	Post at least one paragraph sharing your ReflexModelAgent design in the Vacuum Agent Forum. Post ideas!!! Do not post your code. Use the spellcheck/grammarcheck before posting. Post your chart from step #6.	10	
9.	Read ideas from your peers who created agents using a percept that is different from your assigned percept. Contact at least one peer to discuss a comparison. All involved in a comparison must have completed tasks #1 through #8.		
10.	You and your partner test your ReflexModelAgents on your ten environments designed in task #7 (each of you should have five distinct environments). Post your chart to the forum. 	10	