|  |  |
| --- | --- |
|  | *Vacuum Cleaner World* |
|  | * Introduction   In the book [*Artificial Intelligence: A Modern Approach*](http://aima.cs.berkeley.edu/), Chapter 2 (Intelligent Agents) takes a vacuum-cleaner world to describe the concept of agent function, agent program, and agent designs. In the simple world, the vacuum cleaner agent has a location sensor and a dirt sensor so that it knows where it is (room A or room B) and whether the room is dirty. It can go left, go right, suck, and idle. A possible performance measure is to maximize the number of clean rooms over a certain period.  https://web.ntnu.edu.tw/~tcchiang/ai/Vacuum%20Cleaner%20World.gif  *Artificial Intelligence: A Modern Approach (2nd ed.), Figure 2.2* |
|  | * Extension   In the exercises in Chapter 2, a modified version of the vacuum cleaner world is mentioned. (1) The geography of the environment is unknown. (2) The location sensor is replaced by a bump sensor that detects the agent's attempt to move into an obstacle or to cross the wall. (3) At each time step, each room has a certain chance of increasing 1 unit of dirt.  Here we provide a small program for who are interested in designing agents in the above world. The prior knowledge of the environment and the performance measure are described as follows:  Prior knowledge  (1) The environment is a 10x10 maze. (A 8x8 maze surrounding by walls. See the screenshot at the end of this page.) (2) Each cell is either a wall or a room. (3) The walls are always clean. (4) The agent cannot pass through the wall. (5) The agent can go north, south, east, and west. Each move costs 1 point of energy. (6) The agent can suck, each time decreasing 1 unit of dirt. Each suck costs 2 point of energy. (7) The agent can stay idle, costing no energy.  Performance measure  Given a period *T*, the goal is (1) minimize the sum of square of amount of dirt in all rooms over *T* (2) minimize the consumed energy |
|  | * Source code   There are five modules, agent, environment, evaluator, random number generator, and GUI.  \* agent.cpp/.h implements the design of agent. The main functions are Perceive() and Think(). \* environment.cpp/.h implements the world. The main functions include Change() and AcceptAction(). \* evaluator.cpp/.h implements the performance measure. \* random\_num\_generator.cpp/.h saves a sequence of random values so as to change the amount of dirt in the world without being disturbed by the random actions (if any) of the agent. \* GUI.cpp/.h implements the interface.  To implement your agent, you just need to modify agent.h and agent.cpp by adding data members and/or member functions in class Agent. To have a model-based reflex agent, for example, you may need to add a 2-D array as a data member in Agent to store the maze. You may also need to add some member functions to support the construction of map and then call them in Perceive(). The current version of agent just chooses one action randomly. You can see how it acts and how it performs through the GUI.  [Download the source code](https://web.ntnu.edu.tw/~tcchiang/ai/Agent-GUI-R3a.rar) |
|  | |  |  | | --- | --- | | https://web.ntnu.edu.tw/~tcchiang/ai/Vacuum%20Cleaner%20World%20-%20GUI.jpg  Screenshot of the program | https://web.ntnu.edu.tw/~tcchiang/ai/Vacuum%20Cleaner%20World%20-%20map.jpg  An example of the map file  (1) The first two 5 represent the initial location of the agent in the world. (The green square in the screenshot.)  (2) The value 0.01 means that at each time step the amount of dirty in each room increases by 1 in probability 0.01.  (3) The last value, 1, is a random seed for the change of amount of dirty in the environment. Different agents can be evaluated in a fair way by fixing the value of this seed. | |
|  | * Acknowledgment   Great thanks to Mr. Shi-Yau Yu for his kind help of enhancing my text-version program to a GUI. |
|  | * Contact information   Any suggestion or bug report is welcome to be sent to [me](http://web.ntnu.edu.tw/~tcchiang/). |