Report for Lab 1 in TDDC17, Artificial Intelligence
Intelligent Agents

Margareta Vi
marvi154

Markus Petersson
marpe163

September 6, 2016

1 Algorithm overview

Two different algorithms were used to solve the two different problems. The different solutions are presented in the sections below.

1.1 Task 1 - no obstacles

For task 1, the method chosen was to make the agent go through each row and search for dirt, at the end of each row it turns either to the left och right to move down one row and continues to search for dirt, thus it systematically sweeps every row.

The agent knows when to change direction with the help of a state variable (returning) which is is either -1 or $+1$ to its help. The variable indicates in which direction the agent is heading. To its aid there is also a turning-counter. When bumping, the counter is set to 3 and for each action in a u-turn(right turn, move forward, right turn in our case) the counter is decreased by one. If the turning counter is one and the agents bumps into something again the returning variable is set to the inverse of itself, $+1$ or -1, and the agent changes sweeping direction.

To make sure that is stops at the home position, a counter is used. For each time the home position is passed, it increases the counter with one and at the second time it passes home it stops at that position.

1.2 Task 2 - with obstacles

Task 2 required a little more thought than task 1, since obstacles now are obstructing the agent. The agent bases its decisions on whether adjacent squares are unexplored or not. If there are unexplored squares then one of these must be chosen. The tricky part is what happens when there are no unexplored adjacent squares.

The agent now holds to the left so as long as there is unexplored paths to the left it goes to the left, otherwise it tries to go to forward and lastly to the right (if unexplored). To make sure that the agent does not fall into an infinite loop, there is an randomness in the program, which is triggered when the agent comes to a dead end. Then the program randomizes to either perform a left or right turn for the agent (including a move forward), in order to avoid infinite loops. When the agent encounters previously unexplored squares it returns to it's primary behaviour.

The agents stops when there is no more unexplored squares on the map. It knows that the map is 15x15 squares.

2 Disscusions

The methods chosen are rather primitive, but introduces enough intelligent behaviour to solve the task at hand. There is, however, a lot of room for improvement. E.g. the introduction of a good heuristic search algorithm, such as A^*, would decrease the time to solve the problem tremendously. One thing that is good with the implemented algorithm is that it requires few computational resources.