

CSEN403 PROJECT 2 REPORT

Team Number: 10

-Omar Mohammad Salah / 52-1663 / T-15

-Khaled Mohammed / 52- 2300/ T- 25

-Zeyad Tarek Nabil / 52- 7231/ T-19

-Ziad Ahmed Abdelazim / 52-4345 / T-19

Project description:

We implemented several methods (10) to initiate the AI module for a minesweeper robot in Haskell. First, we identified the type cell referring to the coordinate cells of the robot and the mines to be collected, and the data MyState referring to the robots state up until reaching the goal of collecting all mines in the grid. Moreover, there were four methods implemented to allow movement in the grid and updating the input state (up, down, left, right). The method "collect" checks if the robot is in the same position as one of the mines and removes the coordinates of the mine collected from the array of mines positions. The method "nextMyStates" checks and returns a set of states resulting from applying any of the methods above (up, down, left, right, collect) from the input state. The method "isGoal" which takes as input a state and checks if the mines list is empty. The method "search" which takes as an input a list of states and returns the goal state by calling itself recursively if the head is not a goal state on the tail of the list, and nextMystates of the head state. The method "constructSolution" which outputs a list of strings containing the actions taken by the robot to reach the current state in the input from the initial state. Finally, The method "solve" which takes two inputs, the initial position of the robot and the a list containing the positions of the two mines and returns the a list of strings containing the actions that should be taken by the robot to collect the mines.

Grid 1:

| | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | | | | |
| 1 | | | X | |
| 2 | | | X | |
| 3 | R | | | |

```
Main> solve (3,0) [(2,2),(1,2)]  
["up","right","right","collect","up","collect"]  
Main> |
```

Grid 2:

| | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | R | | | |
| 1 | | | | |
| 2 | | | | |
| 3 | X | | | X |

```
Main> solve (0,0) [(3,0),(3,3)]  
["down","down","down","collect","right","right","right","collect"]  
Main>
```
