Lab1

Teodor Riddarhaage (teori199)
Pontus Svensson (ponsv690)

September 2018

1 The Agent

The agent discovers the world through an exploring algorithm using three flat lists.

- **squareStack** - Holds the squares waiting to be visited.
- **visited** - Holds the squares that have been visited.
- **path** - Holds the path from the starting square to the current square.

The algorithm works as follows:

1. Push the current square's four adjacent squares to *squareStack* unless they are already in *squareStack* or in *visited*.
2. Add the current square to *visited*.
3. Depending on the percept, the agent chooses different actions. If the agent perceives dirt, it will suck. Else the agent will try to move to the next square in *squareStack*.
4. If *squareStack* is not empty and the top square is adjacent to the agent, it will move to that square, pop it from *squareStack* and add it to *path*. Else, if the square is not adjacent, the agent will backtrack through *path* until it is. While the agent backtracks, the squares it visits will be removed from *path*.
5. If *squareStack* is empty, all squares have been visited and it is time to go home. Using BFS, the agent will find the shortest path to the home square and follow it. If *squareStack* is not empty, the algorithm will return to step one.

2 Motivation of the solution

This method was chosen because we knew that all squares had to be visited. Since all squares adjacent to a reachable square are pushed to a stack, it is guaranteed that all reachable squares will be visited. When the stack is empty, it is also known that all squares have been visited and we know that it is time to go home.

3 Alternative solutions

This solution is not very efficient in this case since the agent will backtrack through, potentially a lot of, already visited squares. An alternative solution that would be more efficient would be to use a search algorithm, e.g. BFS, to generate action sequences for the agent to move to the closest unvisited square. This would result in a shorter path for the agent, but it will also result in more computations.