AI and Logic programming report

Chanthru Uthaya-Shankar
Candidate Number: 51722

Implementation

My implementation for navigating a dynamic world involves repeated use of Breadth First Search (BFS) and A-Star (A*) search.

My thought process was to proceed in the same way as I did for my implementation in Task 3 which involved an intial sweep of the board using BFS to locate Oracles and charge points and use these locations to navigate to using A* search. Quickly I realised that this strategy might not be feasible due to the locations of objects changing too frequently.

I switched my strategy to mainly using BFS to locate and jump to the nearest Oracle, and passed the path generated from my BFS predicate to another predicate that would iterate through and move the agent to each position in the path checking that this position was free from a blocked cell. In the case where the path is blocked by a barrier, the predicate would calculate a new path using A* to the destination position, therefore moving around the barrier cell. If the location of the destination object (oracle or charge station) had changed, the agent would also re-calculate the path using A*.

Limitations

One of the limitations is the speed, calculating the path using BFS takes time and if the positions of objects are moving too frequently the agent may not choose the optimum path to the destination at the point the agent starts moving. If objects are moving too quickly it will fail to find objects on occasions due to the object moving into a position that has already been checked on the agenda in my BFS. If at the time of search all objects are blocked the agent will fail to find the identity even though a path may emerge from the moving of a blocked cell. The agent is not very responsive to changes in its environment, whilst testing my implementation I found that my agent frequently failed to find a valid path when the objects are moving too quickly.