Artificial Intelligence Project Three: Report

Bryan Plant
Department of Computer Science
Montana State University, Bozeman

Keely Weisbeck
Department of Computer Science
Montana State University, Bozeman

Spencer Cornish
Department of Computer Science
Montana State University, Bozeman

1. Introduction.

The purpose of this project was to use a first-order logic reasoning system to simulate the game of Wumpus World. The game of Wumpus World is traditionally represented on a 4X4 grid where the agent must use logic and reason navigate the grid to find the gold. Hidden among the square of the grid there is a monster called the Wumpus and pits that the agent can fall into. The game can end in three ways. The first is if the agent finds the gold and makes it back to the start. The second is if the agent enters the same square as a pit, and the last is if the agent enters the same square as the Wumpus. As the agent navigates the grids it can only see the current square it is in but it can smell adjacent squares. When the agent is near a pit it smells a breeze and when an agent is near the Wumpus it smells the Wumpus stench. Using first-order logic and the smells of the grid the agent needs to find the gold and return back to its starting point at [0,0].

For this project, we implemented two types of agents, a cautious agent, and a smart agent, to attempt to solve the puzzle. We anticipate that the cautious agent will be less likely to find a solution to the puzzle and that it will be more prone to fall into a pit or get killed by the Wumpus. We also predict that overall the cautious agent will need to enter more cells than the smart agent.

2. Algorithms and Implementation Strategies

This program was written using JavaScript and the output is displayed on an interactive web interface. Our implementation is able to generate a game board of various sizes. Along with the traditional 4X4 grid size, our program also offers 5X5, 8X8, and 10X10 grids. Each new game board randomly places a Wampus, the gold, and several pits on the grid. The agent always starts each game on the square [0,0]. Both implementations navigate back to the start of the cave using an iterative breadth-first search.

2.1 Cautious

The cautious algorithm only moves forward onto safe tiles and remembers nothing about what it has previously seen. If it keeps turning around on a specific set of tiles for more than the threshold, the agent will gain some confidence and proceed to a potentially unsafe tile. This only occurs if the agent has no other option.

2.2 Smart

The smart algorithm moves to the safest scored node on the edge of the explored map. On every turn, nodes are analyzed to determine which ones are the safest, based on the knowledge base of percepts the agent remembers from visited tiles.

2.3 First Order Logic rules

The logic rules highlighted in this section are the rules we thought to be the most important rules used by each agent. Not every rule implemented in the program is listed in this section.

2.3.1 assumptions

The variables listed below are only used to define the rules in logic and are not necessarily representative of variables used in the code.

- Stench = we smell a wumpus
- Breeze = we feel a breeze
- Turn(direction) = we turn away from the tile we are currently facing in the direction listed in the parentheses (left, right)
    - If there is not a specified direction then there is a 50/50 chance of turning either direction
- isVisited = we have visited the tile we are currently facing before
- Unvisited = we have not visited the tile before
- Neighbors(type) = all surrounding tiles of the tile we are currently in of the specified type listed in parentheses (visited, unvisited)
- Wall = the tile we are currently facing is a wall
- numVisited = the number of times we have visited the tile we are currently facing
- move(direction) = we move in the direction stated within the parentheses (left, right, forward, back)
- isPit = the tile we are currently facing is a pit
- isWumpus = the tile we are currently facing is a Wumpus
- IncreasePit = increase the probability tile we are currently facing is a pit by 0.3
- IncreaseWumpus = increase the probability tile we are currently facing is a Wumpus by 0.3
- increaseDanger = increase the probability that the tile is a danger by 0.2

2.3.2 cautious agent rules

- If we smell the wumpus or feel a breeze, we turn away from the tile
  $$stench \vee breeze \Rightarrow turn()$$
- If we are facing a visited tile and smell the wumpus or feel a breeze, we walk away
  $$isVisited \wedge (stench \vee breeze) \Rightarrow turn()$$
- If we have visited a tile more than five times, move forward
  $$(numVisited \geq 5) \Rightarrow move(forward)$$
- If we are facing a wall, turn left or right
  $$wall \Rightarrow turn()$$

2.3.3 smart agent rules

- For every visited tile surrounding a tile that is breezy, the probability of a pit increases by 0.3
  $$\forall visitedNeighbors(visited) \wedge breeze \Rightarrow increasePit$$
- For every visited tile surrounding a tile that is smelly, the probability of a wumpus is increased by 0.3
  $$\forall visitedNeighbors(visited) \wedge stench \Rightarrow increaseWumpus$$
- For every tile unvisited P(wumpus) and P(pit) add 0.2 to the probability
  $$\forall unvisitedNeighbors(unvisited) \Rightarrow increaseDanger$$

3. Results

Puzzle	Score	The result of the game	Number of unique cells entered	Total number of moves
4X4	-21	Fell in pit	6	21
5X5	956	Won	15	44
8X8	972	Won	16	28
10X10	-46	Fell in pit	17	46

Table 3.1: Performance measures for each game for the cautious agent

Puzzle	Score	The result of the game	Number of unique cells entered	Total number of moves
4X4	-4	Fell in pit	3	4
5X5	980	Won	11	20
8X8	982	Won	14	18
10X10	-15	Fell in pit	10	15

Table 3.2: Performance measures for each game for the smart agent

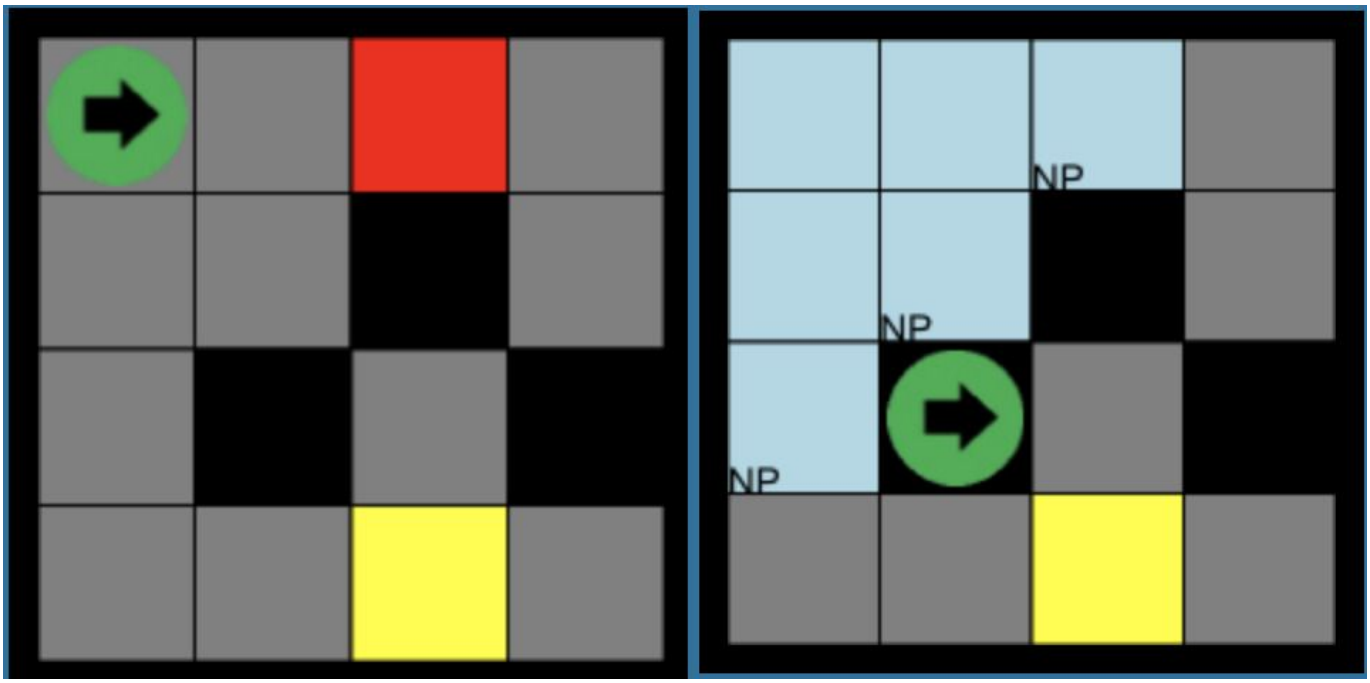

Figure 3.1: Cautious agent's starting and ending solution to the 4X4 grid

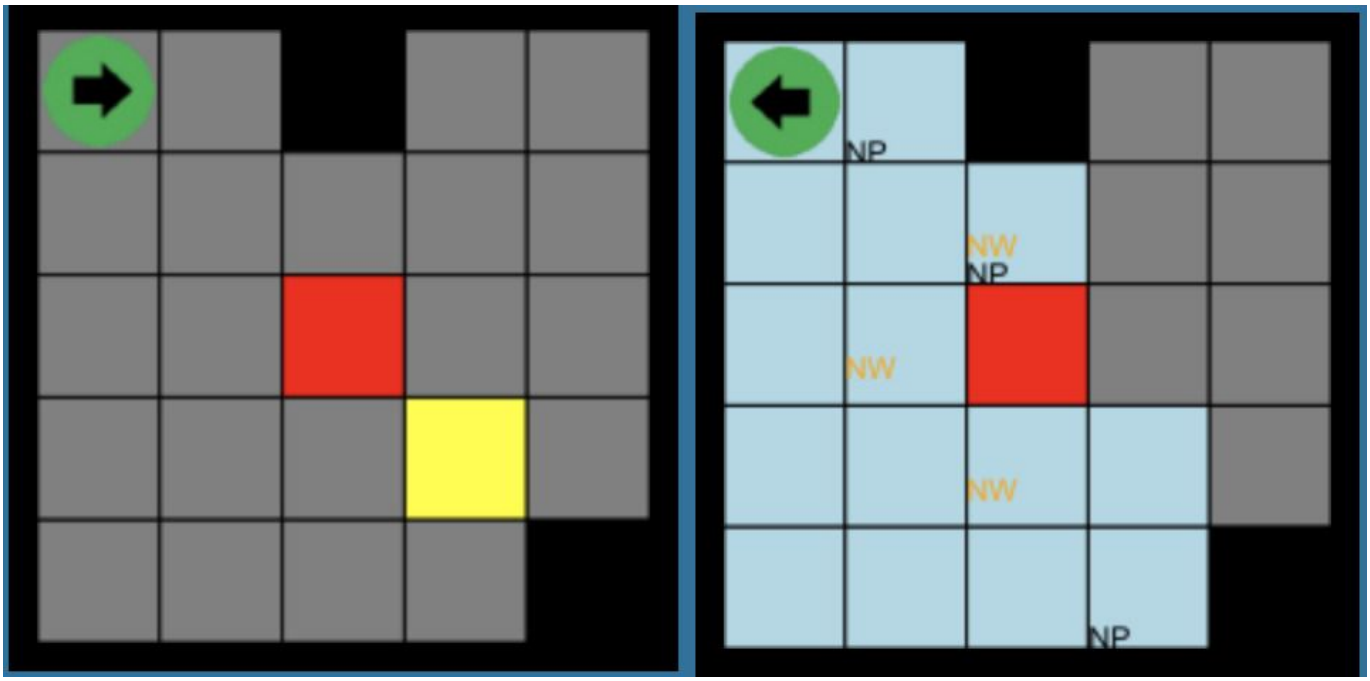

Figure 3.2: Cautious agent's starting and ending solution to the 5X5 grid

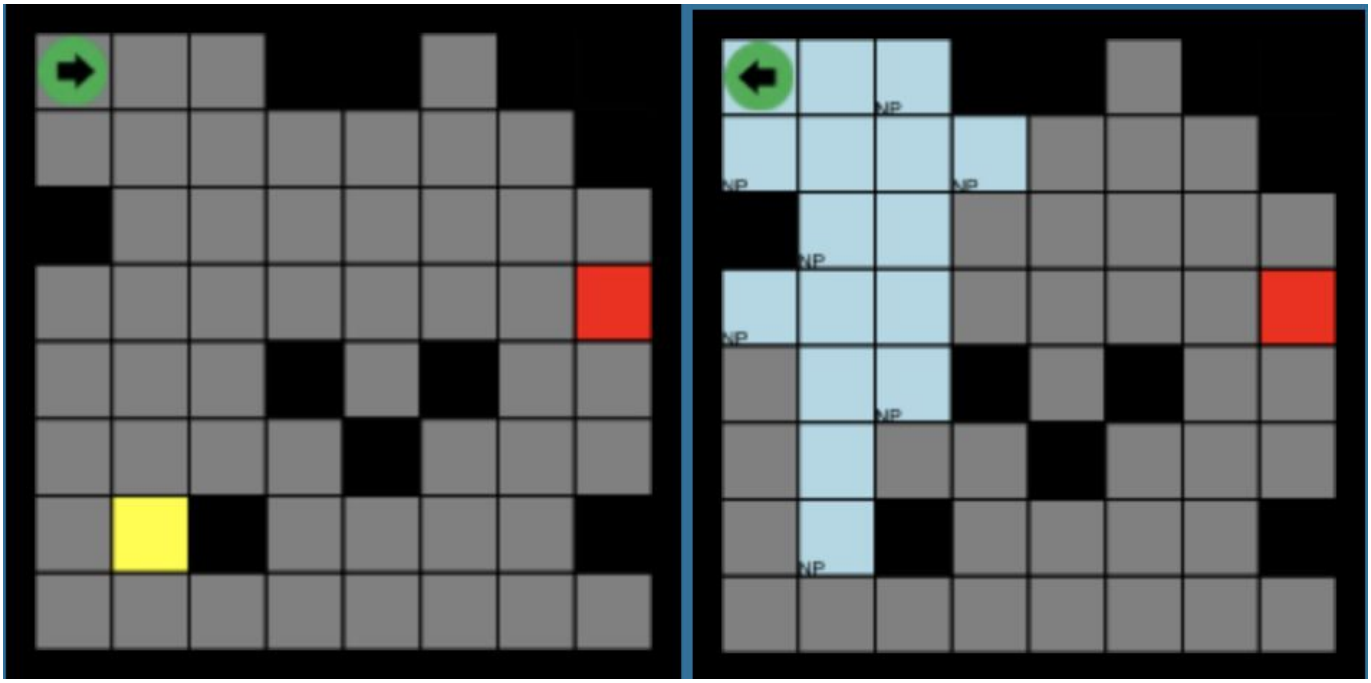

Figure 3.3: Cautious agent's starting and ending solution to the 8X8 grid

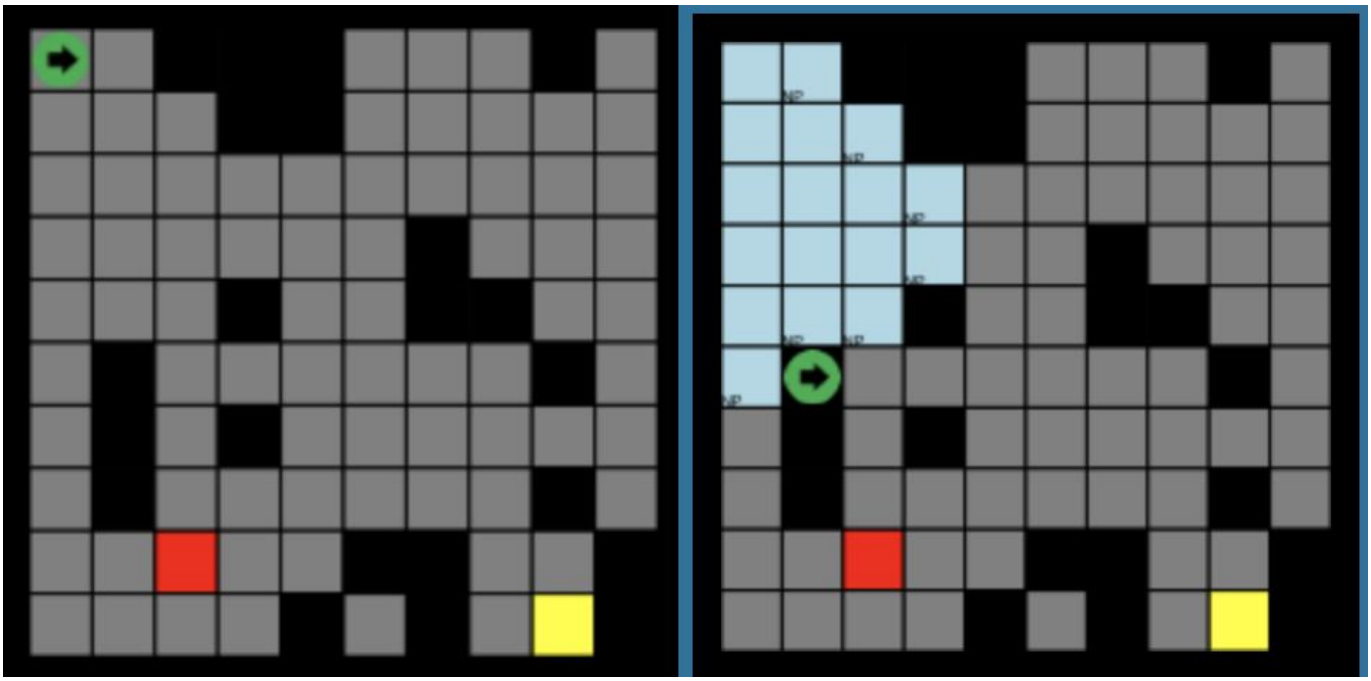

Figure 3.4: Cautious agent's starting and ending solution to the 10X10 grid

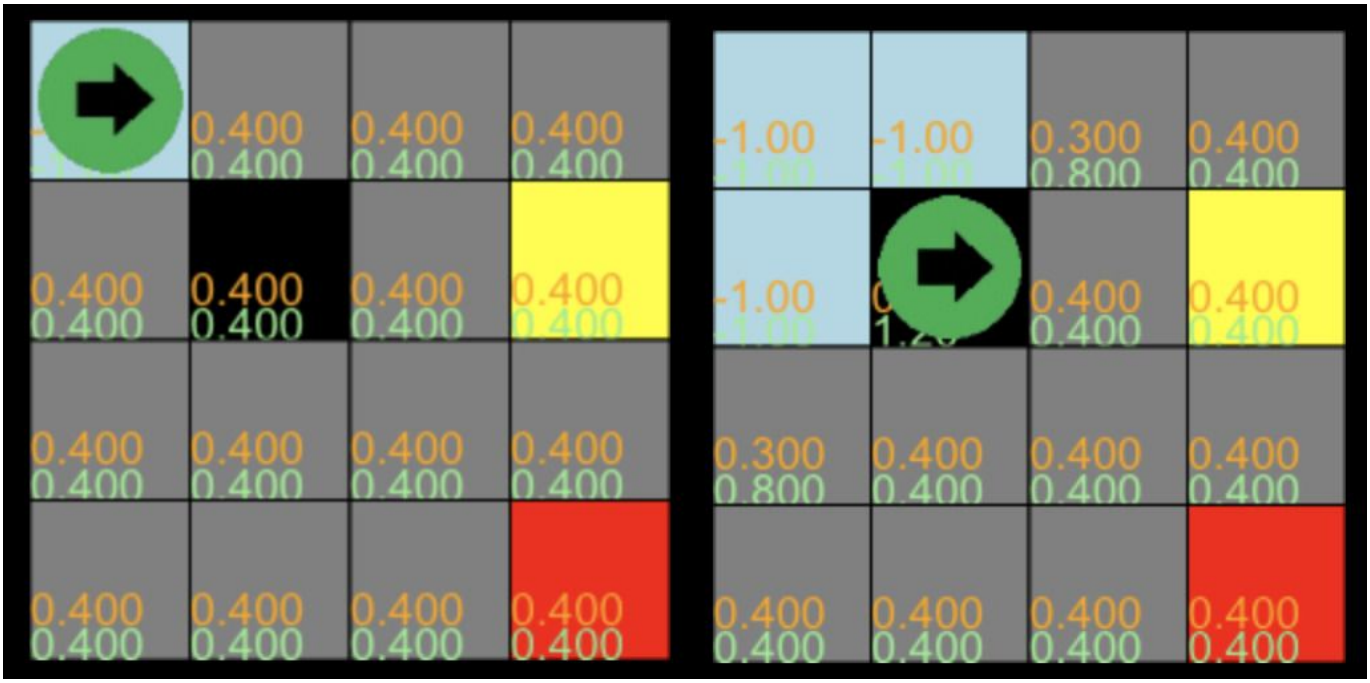

Figure 3.5: Smart agent's starting and ending solution to the 4X4 grid

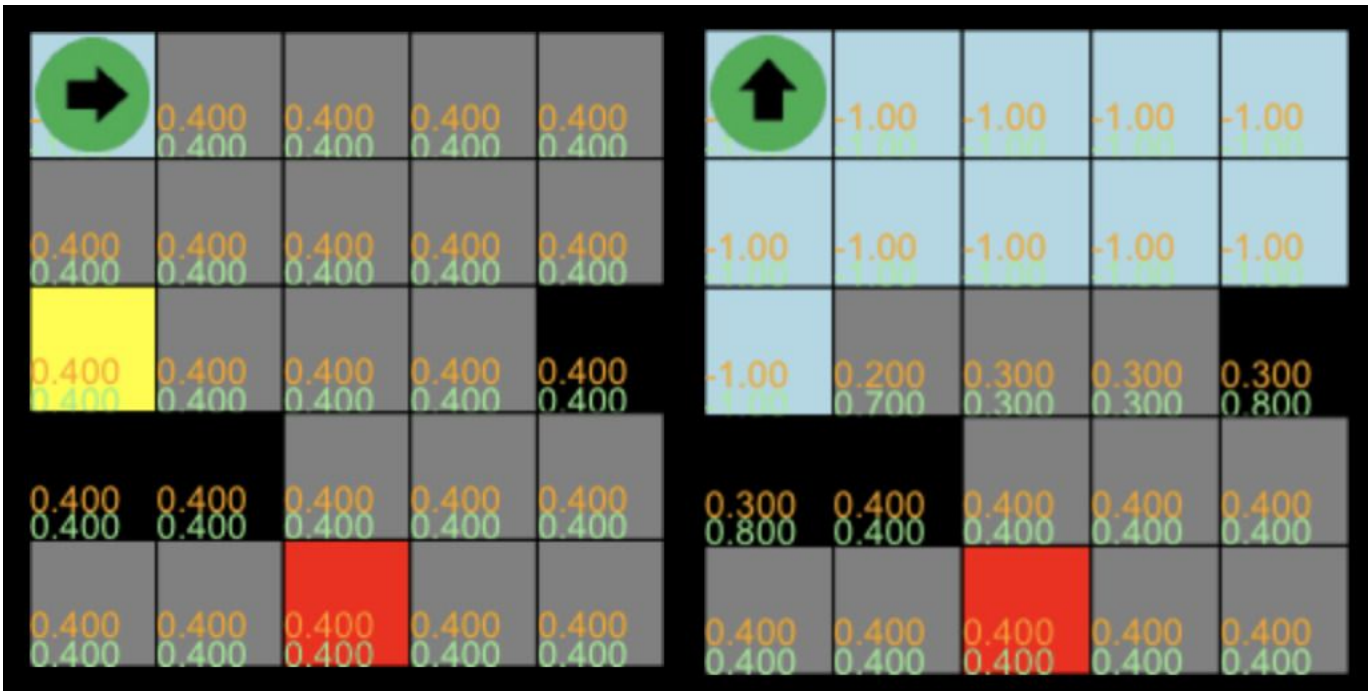

Figure 3.6: Smart agent's starting and ending solution to the 5X5 grid

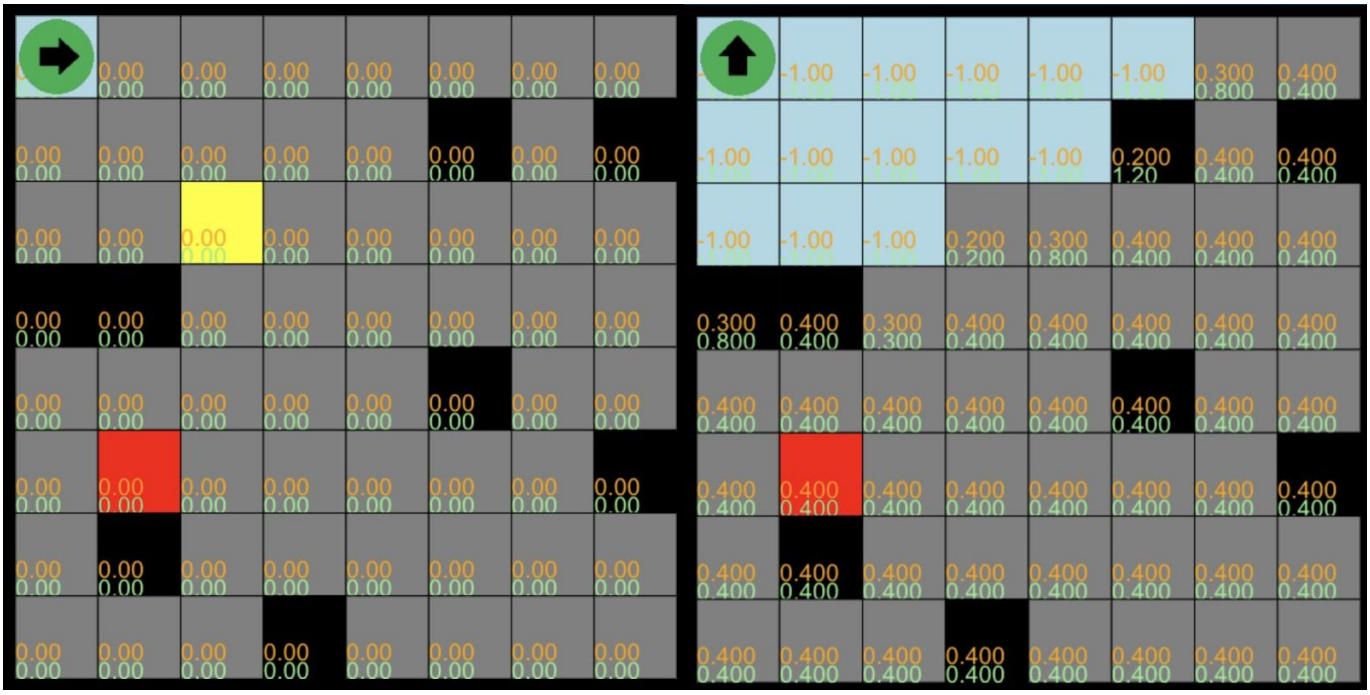

Figure 3.7: Smart agent's starting and ending solution to the 8X8 grid

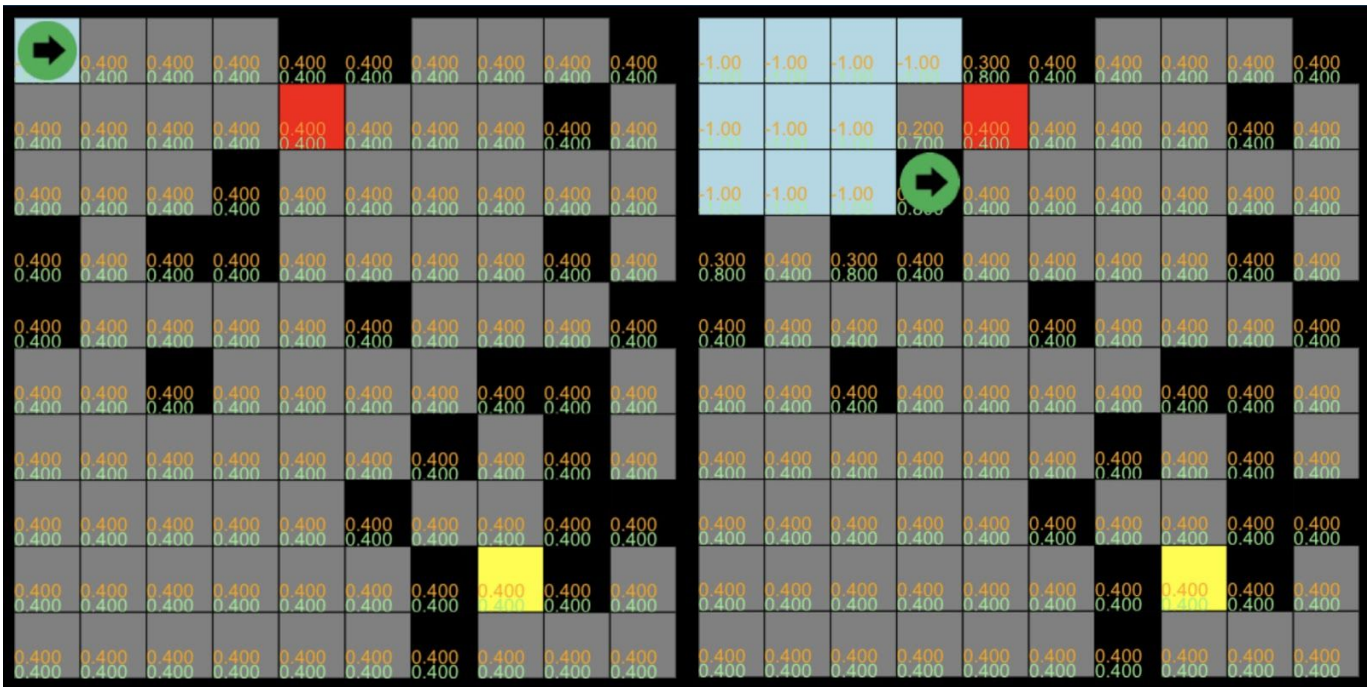

Figure 3.8: Smart agent's starting and ending solution to the 10X10 grid

4. Analysis of Results

We analyzed both agents using three main performance measures: score, the result of the game, and the number of cells entered. The score is set to zero at the beginning of each game. An agent loses one point for every step taken, and ten points if they use their arrow. 1000 points are gained if the agent makes it back to the starting point with the gold. 1000 points are lost if the agent is killed by either the Wumpus or by falling into a pit. The second performance measure we used is the result of the game which can be one of three things. The first is 'Won' where the agent makes it back to the start with the gold. The second is 'Fell in pit' where the agent dies by falling into a pit. The last option is 'Killed by Wumpus' where the agent failed to kill the Wumpus and entered the same square as the monster. The last performance measure used is the number of cells entered, which represents the total number of steps an agent took in that game. Using these performance measures we expected our smart implementation to outperform the cautious attempt. We were surprised when our results did not reflect our initial assumptions.

The most surprising aspect of this data was that the cautious agent sometimes made more progress on a puzzle that the smart agent before losing the game. As can be seen in *Figure 3.1* and *Figure 3.5* as well as *Figure 3.4* and *Figure 3.8,* The cautious agent was able to explore more of the grid before falling into a pit. The smart agent tended to be more reckless in its moves. This could likely be remedied by adjusting the weights of the tile prediction probabilities, as they currently do not perfectly represent the real probability of the tile. For example, if a wumpus is smelled on one side of the map, the probability of a wumpus anywhere other than the neighbors of the smell should be zero. Currently, this is not the case. With more time, we believe that we could adjust the smart implementation such that it could outperform the cautious one by a strong margin.

For the 5X5 and 8X8 grids the results of our tests were as expected, with the smart implementation having made fewer overall moves and entering fewer unique squares to solve the puzzle. As can be seen in *Table 3.1* and *Table 3.2* for both the 5X5 and 8X8 grids the smart agent was able to solve the puzzle in less moves resulting in a better overall score.

5. Contributions

As a team, we walked through several variations of the Wumpus World game on a whiteboard to determine what information the agent required to solve the puzzle. After deciding the most important rules we discussed and worked on implementing them into our first-order logic reasoning system. Each team member contributed to the writing of the final report.

5.1 Bryan Plant:

Bryan was in charge of the initial set up of the wumpus world simulations and helped implement the logic behind the agents.

5.2 Keely Weisbeck:

Keely helped implement the logic behind the agents.

5.3 Spencer Cornish:

Spencer wrote the logic behind the cautious agent and helped implement the logic behind the smart agent. Spencer also submitted the assignment to D2L.