CS-171 Wumpus World Final AI Report

Team name __________ KillWumpus __________

Member #1 (name/id) ___ Shah Masood ___ Member #2 (name/id or N/A) ___ N/A ___

I. In about 1/2 page of text, describe what you did to make your Final AI agent "smart."

I have a lot of data stored in MyAI: a list for potential pits, potential spots for Wumpus, safe zones, visited, current position, carrying gold, previous position, holding arrow, size of board (calculated once hit a bump). Using Arc consistency, MyAI eliminates potential pits if no breezes are found and only traverses safe zones. If the AI visits all the safe zones and if no more safe zones can be created through Arc consistency, or by killing the Wumpus, then it calculates shortest path using BFS to (1,1). Furthermore, the AI only moves to nearest safe square and does not repeat visiting squares unless all the adjacent squares are dangerous or visited. If so, then it uses a heuristic function (distance formula) to find the closest unvisited safe square that uses BFS to find shortest path to that square. In case of tie, it chooses the highest x, y value square because it is the furthest from (1,1). The AI also calculates size of x and y individually once it hits a bump in x axis or y axis, after that, it never hits a bump again in its respective axis. Also, certain tweaks were made such as if the AI is in the top left corner or on the left wall, make it turn right if its facing left (it defaults turning left otherwise).

II. In about 1/4 page of text, describe problems you encountered and how you solved them.

The main problem was of course finding the safe zones which I solved by checking of my safe zones and potential pits by checking if their arc consistency after every move. As I was writing my code, I had trouble with organizing my code and prioritizing what to check for first. For example, my #1 priority was checking if there is glitter and if there is, force my character to go back to (1,1) after grabbing the gold. However, I did not set my characters position since I had organized my code to grab the gold immediately without setting its x and y coordinates. So, my AI assumed it was at a different square and messed up the BFS algorithm. I had issues deciding whether to shoot the Wumpus or not. After several test cases, I found out that allowing it to only shoot when the AI is standing at position less than 3,3 is best because at a large grid, such as a 7x7 grid, there are 49 squares and shooting the arrow just to open up 1-3 squares is not worth it. I even tried waiting until I pinpoint where Wumpus is so I can shoot him but that required a lot of moves being wasted and had even worse outcome than just ignoring it because of moves waste and -10 from shooting the arrow. I found the middle ground being only shooting when AI is at 3,3 because the sizes range from 4-7 and the less the size the more value each square gets. Therefore, if the AI is at higher than 3,3 it ignores Wumpus and explores other safe zones.

III. In about 1/4 page of text, provide suggestions for improving this project.

I feel like the 20% pit spawning leaves it to luck when it comes to probabilistic ways of traversing the cave. It is really risky to take that chance because falling into pit is -1000 points whereas successfully guessing a safe zone next to a breeze is -1 point, a CHANCE (low one at that) of getting gold, and opens up 1-3 square(s). The expected value is way too low. If falling into a pit gave -500 points, I feel like people would actually go into looking at the probabilities of these events because the expected value will be much higher. Another idea would be to have no pit and only multiple Wumpuses + same amound of arrows. This way, we have to be smart about using arrows or occasionally guess where to shoot using probability. Also, having the pits be like minesweeper and tell us the # of pits seems like a good idea as well.