CS-171 Wumpus World Final AI Report

Team name adogisasb

Member #1 (name/id) Tao Wang / 76254921 Member #2 (name/id or N/A) Lingzhe Kong / 93985614

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

First, we set a "safelist" to generator coordinates of safe positions. Originally, the only coordinate in the safelist is the agent's origin which is [0,0]. By a finding neighbor function we can get origin's neighbors as safe moves if the origin is not a breeze nor a stench. We choose a safe position in the list and then move to that position to see if there is any trigger on that position or not. If the new position (new position: a position that the Agent hasn't been there in the history) is still a blank position, the safelist will be enlarged by adding coordinates around the current position and new position will be chosen from the new list. If gold is met at any position on the board, the Agent will grab it and a function that generates the shortest safe path (by Dijkstra algorithm) to the origin will be called and the Agent will go back to the origin immediately. If the safelist is empty when the Agent is at certain blank position (which is really rare), the shortest-path-go-home function will also be called. If the Agent is at a position which is breeze, we will let the Agent go to another safe position in the safelist. If there is no more safe position, the Agent will simply go back to the last position in the moved history. After that, new position will be chosen and the game will go on. If the position is stench, the arrow will be shot immediately to ensure the position next to this one is safe. If scream is detected after shooting, all positions which are stench will be seen as blank positions in the rest of the game. Otherwise, there will be no more shooting and the stench positions will be seen as breeze positions in the rest of game to ensure the safety of the Agent. The part that makes our agent smart is that it will explore the next moves by gathering messages from previous states. By this, our agent can explore more moves, and increasing the opportunity to go to the positions of gold.

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

The most difficult part is we found that implementing the ideas is not as simple as we thought. Covering all possible cases is really hard. Sometime the code crashes at some extremely rare cases but they do exist when testing for 10000 worlds. So, we need to use "-d" for hundreds of times to find out what exactly the cases are. After debugging for the crashes, we used same technique to get the board of the games whose score is lower than -1000 (i.e. the Agent cannot find a way home). This can be the biggest problem that we met in the process of writing this project. Another problem is that we did not prepare a plan of the project at the beginning of the process of coding. So, the whole coding process just became a long journey of fixing this and that. If we did some detailed analysis, such as drew a graph to cover all cases, I think that our programming process will be smoother and we will spend more time on pursuing higher score instead of fixing the bugs of extremely rare cases.

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

The debugging process will be simplified a lot if a set of worlds with all kinds of weird rare cases can be offered as part of the original file of the project. This will save a lot of time that wasted on repeated running to get the extremely rare cases. Since the aim of this project is writing an AI by ourselves instead of experiencing the tedious process of testing, I think such a set of worlds should be provided to help the students ensure their code can cover all the possibilities. Also, I think it would be good if more checkpoints of the project can be set in the whole quarter, such as setting weekly submissions. This will help the TAs and Project administrator know about where is the current position of most of the students and what's the common problems. If those feedback can be offered in the lectures, students will get a chance to know where they are in the whole class as well as some problems that they might meet in the future.