CS-171 Wumpus World Final AI Report

Team name ThunderBall

Member #1 (name/id) Matt Yang/38794873 Member #2 (name/id or N/A) Yi Wang/67023875

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

- We have created a 2D map (11x11) of object called Cell, which models the individual spot in the world and contains wumpus label (initial: int 0), pit label (initial: int 0), orientation (NSWE) availability, and visited label. We also have a position(x,y) to represent the current position of our agent.
- The domain of wumpus label and pit label is $[-1, \infty)$. While exploring the world, if our agent does not feel STENCH, it sets all the wumpus labels of its neighbor cells to -1 (we named -1 as "completely safe", meaning that we are 100% sure that there is no wumpus in that cell, and once the wumpus label has been set to -1, it cannot be changed again).
- If our agent does not feel BREEZE, it sets all the pit label of its neighbor cells to -1 (same usage as what we did for wumpus label). Our agent increments the wumpus label around it by 1 once it feels Stench.
- When any wumpus label >= 2, our agent knows the exact location of the wumpus and will shoot the wumpus. Our agent also increments the pit label around it when it feels BREEZE.
- When the pit label >= 1, our agent knows that there is a possibility of having a pit in that location. Our agent will try to avoid that suspicious location, unless the pit label of that location is set to -1 later once our agent determines the location is pit free.
- The orientation availability is used to determine where the walls are at. Initially, all the orientation availabilities for each cell are true, meaning our agent thinks that there are no walls. When our agent feels BUMP, it knows the boundary of the world depending on where it is facing and will set the orientation availability in all cells with the same x XOR y to false, representing there is a wall.
- The visited label is set to true whenever our agent visits the cell. If our agent finds the gold, it will try to find the optimal path to home based on visited cells. If our agent determines that it is not possible to leave with the gold, it will leave the cave through the optimal escape route it found.

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

Problem: Cannot correctly determine the location of wumpus and pit.

- Solution: Cell with wumpus label and pit label.

Problem: Cost too much when choosing which directions to go.

- Solution: Choose the direction with the least amount of turning.

Problem: When leaving the cave, it is not efficient to use the same path our agent came in from.

- Solution: Calculate the optimal path when the agent decides to leave the cave.

Problem: Cannot determine sequential actions such as move north, move east, move south, move west, move back.

- Solution: Integrate a set of actions to be a movement and push the set of actions into a queue, and then execute all the actions in the queue sequentially (FIFO).

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

- When compiling the file, "make" and "Wumpus_World" are not in the same directory. It takes a lot of time to change back and forth between the directories. When running the world with command –d, it does not show any "std::cout" from the code, which makes it inconvenient when it comes to debugging.

- Don't let the gold be in the pit.