CS-171 Wumpus World Final AI Report

Team name: AZ

Member #1 Thomas Huang / 26768493 **Member #2** Julia Lau / 39121100

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

To make our Final AI smart, we first decided to use constraint satisfaction problems, also known as CSPs, to detect "dangerous zones" and solve the detecting and decision making portion of our AI. Taking advantage of our knowledge of the game rules, the AI can use CSPs to accurately deduce any possible pits and wumpus and act appropriately to avoid dying. Using that same knowledge, the AI can also choose to shoot and kill the Wumpus if the circumstances deemed it necessary. Next, to make our AI move smarter, we taught our AI to seek the shortest path to any non-visited tiles. This was made possible using A* Search with a linked-list implementation. Due to personal design, we gave every tile the same weight when taking into consideration of choosing the next visiting tile since the gold have the same probability of appear on any space regardless of its location. Therefore, the AI will always be choosing the closest unvisited tile, slightly favoring tiles closer to the top and right side of the board. Another one of our design decision is a movement queue. This was created in order to simplify calculations and improve runtime and was made possible because our AI uses a goal based design to move around the map. Therefore our AI can find the nearest tile and the shortest path to the tile in a single calculation.

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

The biggest problem we ran into was having too many ideas, narrowing them down, and making sure all of them work coherently. In order to successfully filter out the redundant ideas, constant revisions of the code were necessary. Although it was tedious and, occasionally, depressing because much of the code were removed. At the same time, the AI grew smarter little by little. Before adding new features, we also made test cases for each individual features to ensure that they worked as intended. However, further modifications were usually necessary after incorporating features into the main code.

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

Personally, I would enjoy the project more if there were more variables we could consider. A few variations of this project could include the getAction function having an input parameters of arrays, which could also give information such as multiple breezes originating from multiple pits.