

CS-171 Wumpus World Final AI Report

Team name Chayuso

Member #1 (name/id) Christopher Ayuso 27096088 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.”

For the minimal AI, I first programmed the AI to simply move forwards until it perceived a stench, breeze, or bump. While exploring, it keeps track of its history tiles and safe tiles, any tiles that don't have a perception mentioned above has the AI presume the adjacent tiles are also “safe” to explore and adds it to the list. If the AI does perceive something besides, glitter to pick up and return, it would create a optimize path to (1,1) from safe tiles.

The draft AI improved its search by not only moving forward, but rerouting its path if it perceived danger or bump. The rerouting would have the AI route an optimized path to the last safe tile it added to its history of unexplored tiles. If the AI found no gold on its journey and all safe tiles had been explored, it would reroute itself to (1,1) to climb out.

The Final AI improved the agent by identifying Wumpus locations from stench tiles and killing the Wumpus if it was safe to explore and was necessary. The AI would then be free to search for gold on the Wumpus improving those worlds that had the gold on the Wumpus.

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

The function I created for finding optimal paths initially would take too much time. I introduced an iterative limit to not exceed searching past n tile length paths. If another path had already been found that reached the tile, the iterative limit would change to that. If still a path was not found, the AI would choose the smallest path closest to the tile to reroute itself.

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

When my AI decides to reroute itself, it chooses the latest unexplored safe tile added to its memory to head to, however, this does not always guarantee that it is the “closest” tile that requires the least moves to head to. I can develop another check to test all unexplored tiles path cost, however, this would only improve the AI by a few points and increase computation time.