CS-171 Wumpus World Final AI Report

Team name__DosVistans____

Member #1(name/id) Jonathan Navarro 83144130 Member #2 (name/id or N/A) Jason Padilla 20502573

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

In order to make our final Wumpus World Agent "smart", we took steps to ensure that the agent would essentially behave as an extremely risk averse individual. Since Wumpus World's design is intent on giving the players the worst possible score, since each move taken is a point taken away from the players score. The very first thing our agent does when a game is initiated is assess current dangers. If our agent is spawned in a tile that has a stench, then it will automatically take steps to locate the Wumpus and kill or avoid it. This is done by shooting an arrow in the "up" direction. If a scream is heard, then we can ignore any subsequent stench percepts because the Wumpus has been killed, and on the chance that no scream is heard, the agent knows that the Wumpus is to its' right. In the case that our Agent is spawned in a world with a breeze in the first tile, then we leave the world entirely in order to not risk death. Once we have passed the first tile, our agent moves around the world by snaking its way through. Essentially, our agent moves up until it hits the ceiling and then moves over to the next column and begins its descent. Once the agent hits the bottom the of the world, then it will move over and up into the next column and repeat this search heuristic. Once it encounters a percept during its search, it alters its course to a diagonal tile. This is done to determine whether the tile that being flanked by possible diagonal percepts is a Pit or Wumpus. If a Wumpus is found, then our agent is instructed to kill it and disregard any further stenches, if the danger is found to be a Pit, then it is skipped over in our search heuristic. The next step that we made in order to ensure that our Agent does not aimlessly wander around (in the case that the gold is in a pit), is that we created a move counter that instructs our agent to leave the world if a certain number of moves has been made. Leaving the world was handled by keeping a list of all visited/safe nodes, and then calculating which visited nodes could lead us back and down to the cave entrance, once our agent was made aware of being back at square one, it would then make the decision to climb out alive and well.

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

One problem that we encountered concerned the fact that our agent used to classify a world as too risky if it was spawned with a danger percept in the very first tile. While this strategy helps avoid death, we found that we were sacrificing too many possible pots of gold by doing so. In order to correct and improve the situation we implemented a strategy for assessing the dangers around the first tile if a percept was present. As mentioned in the previous portion of our report in which we describe the aspects of our agent that make it "smart", the strategy for assessing our first move was actually born from this problem. The basis for fixing solving this problem was to shoot an arrow up if faced with a stench. If a scream was heard, then we can disregard all future stenches. If there was no scream, then we know that the Wumpus is directly to our right and can still disregard future stenches. For breezes, there was no way for us to test the waters for pits, so we still thought that the best course of action would be to leave.Another problem we had concerned the bump percept and correctly keeping track of our coordinates. We were not sure if receiving a bump meant that our agent had run off of the world. In order to fix this issue, we modified the World.py code to print out the game board (for test purposes/undone before submission) in order to see what the AI had done. Upon seeing that the agent stays in place once it bumps into the wall, we looked to our code. Our code updated our AI's coordinates BEFORE returning the our move. For example, if we bumped into the bottom right wall of a 4x4 world, our coordinates would be (4,5) instead of keeping it at (4,4); throwing off any future calculations. The fix was simple as we kept track of our previous moves coordinates and setting our previous coordinates as our current coordinates if a bump was detected.

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

One thing that would be appreciated would be more detailed documentation about how the environment itself works, and certain cases that could arise that were not explicitly stated. For example, mentioning that the gold could be in a tile that also contains a Wumpus/pit would be most helpful for students in the beginning phase of the project. Another suggestion that we feel could help students in the future was if holding a Wumpus specific lecture would be a possibility. The lecture could cover students questions, go over common wumpus strategies/search strategies, and possibly go over the code base in effort to clear up any technical questions that might arise. This could also prove to be a significant time saver in regards to emails that might be sent regarding the same technical questions. One last suggestion for the project, would be to maybe interweave aspects regarding the project into the class lectures more. We personally felt that the lecture in which Wumpus World's were discussed helped a great deal.