Wumpus World Final AI Report

I. Minimal AI

I.A. Briefly describe your Minimal AI algorithm:

Our Minimal AI utilizes the following techniques to obtain a Mean Score of 137.0:

- **Probabilistic Inference:** Let $P(Wumpus_x)$ and $P(Pit_x)$ be the probability of square x having a Wumpus and a Pit, respectively. When visiting x, we set $P(Wumpus_x) = 0$ and $P(Pit_x) = 0$ because we are still alive. On the contrary, if x is not yet visited, then we have to use our historical percepts to infer $P(Wumpus_x)$ and $P(Pit_x)$. If we do not perceive Breeze at x, then we set $P(Pit_y) = 0$, where y is an adjacent square of x; otherwise, we set $P(Pit_y) = 1$ (most conservative estimate). Likewise, we also use the same rationale to deduce $P(Wumpus_y)$ by Stench percept.
- **SAFETY is No.1:** Upon dying, we lose 1000 points, which is not a negligible number. Therefore, no matter what, we only choose 100% safe squares to FORWARD. If there is no unvisited adjacent square which is 100% safe, then we move backwards. Because Wumpus and Pit are independent, for each square x, we use the formula $(1 - P(Wumpus_x))*(1 - P(Pit_x))$ to calculate the probability of being safe at x.
- **Use SHOOT to Query Wumpus:** If Stench is perceived, we use SHOOT to obtain more information about Wumpus. In the case of Scream, we know that Wumpus has died and can set $P(Wumpus_y) = 0$ $\forall$square y. Otherwise, at least we know that there is no Wumpus ahead and can set $P(Wumpus_y) = 0$ $\forall$square y in front of our agent.
- **Set Stop Loss Threshold:** In some cases, it may be impossible or too dangerous to find the gold. In that case, we give up and just go back to the origin. The following are our stop loss thresholds:
  - Have visited the same square more than twice
  - Have visited 80% of squares in the world
- **Prioritize Actions:** Since our goal is to find gold and go back to the origin, we set the following actions as top priorities:
  - If Glitter is perceived, then we immediately GRAB the gold.
  - If we have found the gold or have given up searching and we are at the origin, then we immediately CLIMB.
  - If we perceive Breeze at the origin, then the expected score of FORWARD would be less than ½*expected score if there is a Pit in the front + ½*expected score if there is gold rather than Pit in the front = $1/2*(-1-1000)+1/2*(-1-1-4+1000) = -500.5+497 = -3.5$, so we immediately CLIMB.

I.B. Describe your Minimal AI algorithm's performance:

Cave Size	Sample size	Mean Score	Standard Deviation	99% Confidence Interval
4x4	1,000	229.0	515.6	229.0 ± 42.1
5x5	1,000	157.9	492.8	157.9 ± 40.2
6x6	1,000	100.9	451.0	100.9 ± 36.8
7x7	1,000	60.0	454.5	60.0 ± 37.1
Total Summary	4,000	137.0	479.2	137.0 ± 19.5

II. Draft AI

II.A. Briefly describe your Draft AI algorithm, focusing mainly on the changes since Minimal AI:

Our Draft AI utilizes the following techniques to obtain a Mean Score of 238.0 (+101.0):

- **Depth-first Search (DFS):** Instead of wandering unguided, we systematically push safe and unexplored squares into a DFS queue, and then expand nodes based on it. In that way, we can perform blanket search through the world as much as we can. If the DFS queue becomes empty while we still do not find the gold, it means that we have explored all 100% safe squares and gold may be in a Pit or surrounded by Pits, so it is time to lift a white flag and retreat. (If the DFS queue becomes empty but we have Arrow and know where Wumpus is, then we kill the Wumpus and expand more nodes.)
- **Logical Inference:** If at some point we find that there is only one square x among all adjacent squares of a Stench square which has $P(Wumpus_x) \neq 0$, then we can infer that Wumpus lives at x. If x had no Wumpus, we would not perceive Stench. In that case, we can set $P(Wumpus_y) = 0\ \forall y \neq x$, and we do not have to care about Stench anymore.
- **Random Walk:** If we hard-code moving rules, then in some tricky cases we may get stuck in the same place until losing 1000 points. To solve the problem, inspired by Simulated Annealing, we randomly assign where to go next as long as the next square is safe. For example, if both Up and Right are safe, sometimes our agent may move upwards, and sometimes our agent may move rightwards. We find that the randomization can somehow help our agent escape some infinite loops and move on.

II.B. Describe your Draft AI algorithm's performance:

Cave Size	Sample size	Mean Score	Standard Deviation	99% Confidence Interval
4x4	1,000	335.3	474.3	335.3 ± 38.7
5x5	1,000	257.0	443.4	257.0 ± 36.2
6x6	1,000	201.1	411.5	201.1 ± 33.6
7x7	1,000	158.7	382.7	158.7 ± 31.2
Total Summary	4,000	238.0	429.3	238.0 ± 17.5

III. Final AI

III.A. Briefly describe your Final AI algorithm, focusing mainly on the changes since Draft AI:

Our Final AI utilizes the following technique to obtain a Mean Score of 241.6 (+3.6):

- **Record the Way Home:** Even though we randomly assign where to go next, in some situations our agent still gets stuck in the same place, particularly when retreating to the origin. Thus, we decide to record the path stemming from the origin while our agent is exploring the world. If the way home is rather roundabout, our agent just follows the original path to go back to the origin.

III.B. Describe your Final AI algorithm's performance:

Cave Size	Sample size	Mean Score	Standard Deviation	99% Confidence Interval
4x4	1,000	339.4	471.5	339.4 ± 38.5
5x5	1,000	261.1	441.1	261.1 ± 36.0
6x6	1,000	204.9	409.0	204.9 ± 33.4
7x7	1,000	160.8	381.4	160.8 ± 31.1
Total Summary	4,000	241.6	427.1	241.6 ± 17.4