Me being a human, I would solve the maze by simply trial and error. If I fail, I will go a different way in the maze to finish it. The machine is the same way, in it would try every possible way to finish it, however, would possibly not do so in the quickest way possible. Humans tend to push themselves as hard as they can to see how fast they can do something without taking a step that could lead to failure. However, as we have seen from things like AlphaGo if the product is a win, it does not matter if a step taken could be considered a faulty one. The similarities in the agent we built is a reward system based, in which it learns where to find the treasure after each bad move and then corrects itself. Though it is not as complicated as GO in this circumstance the way the agent is trying to learn the game is like how I would approach the game.

Exploitation and exploration are quite different, exploitation in my opinion is the innovation that comes from already existing methods and then are greatly improved upon. Exploration however is finding a new way to go about a problem to come to a solution that has not been made before. In my opinion, this is the more fun way of programming though it does require a lot of research which can also be fun. You are a pioneer of a new method or way of thinking that could benefit thousands of people from your code or fix issues several people are having. In most cases exploitation is reinventing the same wheel while exploration is making something completely different. I would say humans would use more exploration for this pathfinding problem as they want to solve it the most efficient way possible. However, the algorithm or agent would be more of the exploitation concept here. This is because it simply does not care how fast the problem is solved if it is solved and wins at the end. Reinforcement learning can help determine the path to the goal as it constantly updates the agent with possibly outcomes. In doing so, the agent can get smarter and smarter when it comes to finding the correct path and winning the game.

To solve the treasure hunter game, I implemented deep Q learning with the algorithm. In doing so I created an algorithm that continues and learns from failed mistakes yet rewards the system for correct moves. In doing so, the agent then learns from a series of trial and errors on better moves to make to get rewarded from to speed up the winning process and have a higher chance of winning the treasure hunter game. It does this by comparing past moves to calculate a higher chance of winning through previous move sets.