**Teodoir O’Ceallaigh**

**CS-370**

**Southern New Hampshire University**

**October 2022**

**Project Two - Design Defense**

**Human VS Machine Approach**

When it comes to approaching problem solving, humans and machines function rather differently. In the case of a human approaching this particular problem, the maze, the approach would most likely be a logic series of trial and error with learnings applied immediately as new information is garnered turn by turn. In the case of machine learning, the running of epochs provides a quantified amount of randomly generated scenarios, or paths, that are attempted. Theis series of epochs, or trial sets, would be completed by going through an initial step of ingesting the problem before running through the problem multiple times until the most efficient path is determined. Once that path is determined, or the highest rate of efficiency/success is reached in the number of epochs entered in the algorithm, then that most efficient path is used as the basis for future actioning.

In terms of similarities, both parties take information from the reaction their action caused, ingest that data and utilize it to make the next action. Where there can be a large gap in the two is the instances of reward-based machine learning. Whereas a machine would be driven to ake the most logical course for the highest reward after their action, humans may veer down a different path based on future-thought logic and how the reward could grow or diminish based on foreseen paths.

**Intelligent Agent Pathfinding**

Explotation and exploration differ in the way that they approach resolving the problem. Exploration methods involve searching, testing, locating, and iterating on each and every possible series of solutions. Exploitation methods involve reviewing the entire selection of potential inputs and then further testing every series of solution available. With our current use case the best mix of the two is to utilize the exploitation of the variety of paths while utilizing attempts of exploration mixed in to allow for new information to be ingested by the agent. Reinforcement leaning can heavily support this goal through incentivizing trial and error via exploration as the agent would need to discover new ways of testing for the solution in order to complete the path successfully to the result of a close to, if not a full, 100% success rate.

**Solving Problems With Complex Algorithms**

For this game I implemented deep Q-learning by setting up and utilizing the appropriate library resourcing and environment, curating the reward system, developing the appropriate code for the learning agent in our game, and using and enhancing the algorithm through my own testing for best use. Once iterated I was able to implement the testing with the agent in the Jupyter Notebook. These efforts to utilize a neural network and enhance it through reward systems creates an environment for finding the most efficient and rewarding series of actions to reach he ultimate goal of the treasure cell.