Zachary Meisner  
12/13/22  
Professor Alexander  
Current/Emerging Trends in CS  
Project 2

* **Analyze the differences between human and machine approaches to solving problems.**
  + Describe the steps a human being would take to solve this maze.

The steps needed for a human being to solve this maze are relatively simple if you break it down. Within the area of this 8x8 matrix. If we were to expect any random type of maze, a human that has not done this in any manner would start in an exploration stage. This is the stage in which we are to try and highlight any usable patterns to create an understanding of the type of maze that we have and then make sense of it. This segmentation is important because it is what helps humans understand and process information in the most efficient way. If the human solving this has notable experience, they would understand these patterns, and be able to understand and respond to them in the quickest and most appropriate way by taking percent chances that create what you could consider to be a success rate of each decision. Overall, it would take an incredibly long time for a human to do this, and it likely would take an incredibly long time for any of the learned information to become something that they could do without having to think about it.

* + Describe the steps your intelligent agent is taking to solve this pathfinding problem.

The steps that my intelligent pirate agent is taking to solve this pathfinding problem involve movement, experience, and exploration. Firstly, the agent can move left, right, up, and down. Meaning that we are charting movements imposed onto a 2D matrix. The size of this matrix contributes to the time it takes for the agent to explore, and learn, in addition to the potential complexity of the puzzle which is what our enumeration of educated guesses that our pirate agent ends up using to navigate puzzles essentially is crux of the logic that drives movement. Experience for our intelligent pirate agent is defined by something called epsilon, which in our case is 0.1. This means that for every ten attempts, the agent will attempt to learn by experience nine times, and randomly explore a new path one time. What this does is reinforce the foundation of the values that help teach the agent based on a set of criteria defining positive or negative movements which are primarily driven by attaining the goal of the exercise, which in this case is finding the treasure. Lastly, exploration which pairs similarly with experience and epsilon, it is important to point out that this is experience through exploitation, which is a fundamental concept of artificial intelligence that helps the overall creation of what you could consider to be the network of nodes that drives the application. Exploitation is where you use the hard data that you make an educated data-driven decision to determine the actions that you take which is what makes this such a strong concept when it involves the manipulation of the overall pattern of the puzzle. This, mixed with exploration makes an incredibly strong pair, because now our agent can learn of new approaches and solutions in addition to using methodologies in the past that the agent has learned.

* + What are the similarities and differences between these two approaches?

There are plenty of similarities between AI and human application of problem solving. Mainly because we are incapable as humans to truly deviate from the conceptual problems of thought given that everything in the world, to some degree, is just a re-iteration of something else on a lesser, and even granular level. This recursiveness allows us to identify the number of patterns and applications that we can use in different instances, which is how our intelligent pirate agent is also modeled, as we are considerably trying to simulate what a human could do if their brain had superpowers. One drawback of humans in this instance compared to AI, is that the average person does not have the capability of learning something in such a quick, repetitive manner. On the contrary, when it comes to the application in a game setting, AI can be trained, and have been trained in the past to have the capability of consistently overcoming the best of the best that mankind has to offer, which is not something that a normal human usually has the capability or even the capacity to be able to do.

* **Assess the purpose of the intelligent agent in pathfinding.**
  + What is the difference between exploitation and exploration? What is the ideal proportion of exploitation and exploration for this pathfinding problem? Explain your reasoning.

As mentioned prior, exploration is the choice to make new decisions whereas exploitation is the decision to make decisions based on prior data, vouching for a linear constant of response in a case-by-case scenario within each encapsulated instance of movement and positioning versus the maze. The balance between the two is incredibly important as if we never choose to explore, our agent will not learn new things, and not be driven towards the optimization that it inevitably will continue to grow. On the other hand, if we have no exploitation, there will be no incentive to make any type of informed decision there will be little to no structure for our agent, and there will be a nonsensical pattern of movement executed which will inevitably defeat the entire purpose of using the AI as there will be no optimization. One strategy to help define the perfect balance of our agent is called the epsilon greedy strategy. This is where we start out initially with an epsilon of 1 which would define a state of total exploration and no exploitation. After a baseline is created, there is a gradient descent of the epsilon to 0.9, 0.8 and so on, which changes the ratio of exploitation to exploration to a point where the decisions our agent makes are informed, and data driven to a point in which there is little to no doubt of the winning.

* + How can reinforcement learning help to determine the path to the goal (the treasure) by the agent (the pirate)?

The application of reinforcement learning to pathfinding helps our pirate find the treasure in a superior way due to the amorphous state of being that the iteration of each maze can hold. Thinking from a smaller frame of view, we could assume that in the case that you were an actual pirate, trying to find your way through a maze, you could subject it down to the four directional pathways and the existence, or non-existence of these pathways and the actions you could take when given the chance for you to make a decision of some kind on which way to advance. Studying the potential underlying biases and patterns when it comes to the computer-generated maze has quite a lot of implications past the initial 2D matrix. I mean this in reference to understanding the linear state of being that everything in the world displays, in addition to the multi-dimensional state of outcome and chance that someone could theoretically play with when it comes down to the decision and logic implemented into the initial creation of the maze. Adding more features and components to this maze can give us the ability to study real life phenomena and instances in which could help real life people, for example, if we sent in a robot to save someone and used matrix factorization and image recognition to feed into the Q-Learning network to help find someone that maybe a victim of natural disaster. Understanding the conciseness of the application in this instance truly enlightens the instances in which we see how reinforcement learning helps determine pathfinding, and this is just at the beginning stages.

* **Evaluate the use of algorithms to solve complex problems.**
  + How did you implement deep Q-learning using neural networks for this game?

Deep Q-learning uses Experience Replay to learn from epochs which represent the contextual state of outcome and measures the actions our AI agent has taken against the “rewards” that it achieved. The multitude of states and the rate of change between each state helps the AI measure the overall success and pattern rate and pairs that to specified actions taken and whichever scenario given. In our instance, we are using the state of each epoch to measure the success and failure rate of each decision made that is encapsulated within the multitude of patterns that come into existence against the randomly generated mazes. This is allocated with a specified amount of memory in addition to variables that subject the state of change for our agent to a multitude of continually ongoing instances created by the developer. Overall, it shows you the depth of ability we have to manipulate the overall application of the state of change and how the differing outcomes of each variable affect the overall performance.