**Analyze the differences between human and machine approaches to solving problems**.

Humans have a myriad of ways to solve problems based on their past experiences, in this way they can be similar to AI. So how a human would solve the maze would depend on what those experiences were, similar to how an AI would draw on its previous episodes or decide based on an equation it may have been evolving for the course of its training. Me personally, I have always used the left-hand rule to solve mazes, where you either place or imagine placing your left hand on the wall of the maze and never let go. This runs into issues when you encounter islands, portions of maze disconnected from any exterior wall, but a little bit of improvisation and using other external clues can keep you on course. Granted this method is not quick or efficient, it will eventually get you out of the maze. As a human, I can also decide to only do mazes when time is not a factor, so I’m not punished for such inefficiency. If I had an arial view or all the information of the maze at once, I typically solve a bit from the start, then the end, and alternate back and forth until they meet.

An AI would consider a similar, but much smaller, set of factors when attempting to solve this maze. A lot of it comes down to how you as a developer want to write the policy that the AI uses. How much do you want to discount future information, how much do you the AI to explore or exploit, what kind of algorithm do you implement for the AI to follow? Our pirate agent explored by looking at the tiles it could move to next, and possibly the next couple of moves, and deciding which would yield the best outcome. This way we can have the AI consider the future options so it can “calculate the maximum expected future rewards for action at each state” (Lamba), this is a form of Q-learning. In a way, the AI operates similarly to how I described my decision making for solving a maze with the left-hand rule. The factors I consider to be important are often not related to time, but completing the maze, and my past experience tells me this method works with those parameters. The AI doesn’t get to explicitly choose the variables it values, but it does have a list of factors that are calculated into its past experiences and makes informed decisions based on those.

**Assess the purpose of the intelligent agent in pathfinding**.

In the context of this pathfinding problem, exploitation would be choosing the path that seems best based on the current information or policy. In our specific case, we focused primarily on the ability to complete the maze with little regard for efficiency of moves, so using any prior knowledge on how to complete the course would be exploitation. Exploration in this context would be the agent trying to find faster or alternative routes to the finish. Depending on the type of model your AI is using, this may be used to update your policy in varying ways reflected in that model. A quick way I have tried to memorize these differences is that the agent *exploits* past information but *explores* new information. As a pneumonic this can be deconstructed pretty easily, but it works for me as I currently understand these two concepts.

Balancing exploration and exploitation can be challenging and vary depending on your resources and constraints, one group of researchers wrote “The challenge for the agent… is to balance exploration of the two alternatives with exploitation of information gained so far” (Botvinik). The balance between these two concepts, in my mind, is predicated on the time constraints. As the balance can be affected by the learning rate, the size of the maze, the complexity of the maze, and a couple other factors. But all of these factors impact the time the AI takes to learn/master/complete the task you set it. So in our case, the maze was small and simple and the time constraint was technically 1 week. This means we can afford to do some more exploration, which may take longer, but yield a more optimized result.

From here we can look at how all of this can be used to apply the concepts of reinforcement learning to our agent. As with all topics regarding AI and their learning, there are many ways to approach a problem and utilize the different tools presented. One way discussed in our readings would be to use an epsilon strategy, that is, “The [agent] will explore the environment and randomly choose actions… As the robot explores the environment, the epsilon rate decreases and the robot starts to exploit the environment” (Lamba), this allows the robot to gather more knowledge early and then use that wealth later to exploit and extrapolate values. Another option would be to look at gradients and gradient policies. This method of reinforcement learning allows our pirate agent to update its policy within the gradient while learning in efforts to reduce falling into local minimums. However, there is an potential issue of “noisy gradients and high variance” (Yoon) causing either longer learning periods, or less optimized solutions if not given enough time.

**Evaluate the use of algorithms to solve complex problems**.

For this project in particular, we were focused on creating a simple agent and experimenting with some of the variables to see how it influenced the length of learning and the quality of the results. We designed our agent to attempt the maze within a given number of moves and episodes, learning from each as it went along. Occasionally the agent would use a previous attempt (episode) to learn from and possibly update its approach. The agent uses its neural network to map values to moves and come up with a solution to the maze, then to optimize that solution.

Sources:

Botvinick, Matthew, et al. “Reinforcement learning, fast and slow.” Trends in Cognitive Sciences, vol. 23, no. 5, 2019, pp. 408–422, https://doi.org/10.1016/j.tics.2019.02.006.

Lamba, Akshay. “An Introduction to Q-Learning: Reinforcement Learning.” *Medium*, We’ve moved to freeCodeCamp.org/news, 3 Sept. 2018, medium.com/free-code-camp/an-introduction-to-q-learning-reinforcement-learning-14ac0b4493cc.

Yoon, Chris. “Understanding Actor Critic Methods.” *Medium*, Towards Data Science, 17 July 2019, towardsdatascience.com/understanding-actor-critic-methods-931b97b6df3f.