First, the environment, which is a maze represented by a NumPy array, must be defined. Ones indicate open roads in this array, whereas zeros indicate obstructions. The intelligent agent engages with the surroundings, making an effort to navigate the labyrinth and reach the prize, which is situated at a certain spot. A function that shows the maze and highlights the agent's current position, visited cells, and prize location is used to visualize the agent's progress(Xu).

The agent use an exploration factor called epsilon to balance its behavior between exploring new paths and taking advantage of well-trod ones (Hilda). It makes decisions based on a set of predetermined actions (going left, right, up, or down). By choosing a course of action and seeing what happens—including any rewards it may earn and if the game ends—the agent engages with the environment (whether it reaches the prize or gets stuck).

A key component of the agent's learning process is the application of deep Q-learning. Here, the Q-values—which stand for the predicted future rewards for doing particular actions in particular states—are approximated using a neural network(Zhu). The agent continually improves its Q-value estimations to make better decisions as it navigates the maze and gains experience. The agent learns steadily and effectively because the neural network is taught using events that are kept in a replay buffer.

There are some parallels and variances between the methods used by people and robots to solve this labyrinth issue. When a human were to solve the labyrinth, they would start by taking in the layout of the maze visually(Druga). They would mentally map out a route based on this visual information, usually choosing the one that seemed to be the safest or shortest. They would use their cognitive abilities to adapt and find their way through the maze by changing their plan of attack in response to impediments.

The machine, however, functions in a distinct way. The intelligent agent's capacity for cognitive mapping differs from that of a human. Rather, it uses trial and error to explore the maze, first attempting a variety of pathways without settling on one(Yeap). The agent gradually learns which pathways are more promising as it receives feedback in the form of incentives (for approaching the prize) or penalties (for running into barriers). The agent's Q-values, which serve as a measure of this learning, direct its subsequent decisions. Both people and robots use experience to modify their techniques, although humans primarily rely on instantaneous visual and cognitive processing, while computers use mathematical models and data gathered from several trials.

In pathfinding, the intelligent agent's goal is to locate the quickest way to the prize by weighing the benefits of using well-traveled roads against those that are less well-traveled. Exploration means taking new, uncharted roads and trying them out; this might lead to the discovery of quicker or better routes. On the other hand, exploitation entails taking well-traveled routes that have produced positive outcomes in the past. The agent may concentrate more on exploration in the early phases of learning in order to obtain enough knowledge about the maze. It turns toward exploitation as it gains knowledge and becomes more adept at navigating the maze; based on prior experiences, it favors routes that are most likely to lead to the wealth.

One important component of this process is reinforcement learning. The agent has the ability to correlate particular behaviors with favorable results (getting closer to the prize) and unfavorable results (running into obstacles or turning away from the treasure) through reinforcement learning. The agent wants to locate the quickest or most effective route to the treasure in order to maximize its cumulative reward (Ruiz). The agent uses neural networks to approximate the Q-values using deep Q-learning, which helps it decide which actions have the highest chance of success.

For this pathfinding issue, the neural network approximates the predicted future rewards for every action that might be taken in a given state by using deep Q-learning. This enables the agent to assess several routes and determine the best course of action to optimize its chances of locating the prize. Replay buffers containing historical data are used to train neural networks. The agent gains the ability to extrapolate from previous experiences to novel, unseen scenarios in the labyrinth, which strengthens and improves its decision-making stability with repeated training on these encounters.

Druga, S., Williams, R., Park, H. W., & Breazeal, C. (2018). How smart are the smart toys? In Proceedings of the 17th ACM Conference on Interaction Design and Children. IDC ’18: Interaction Design and Children. ACM. https://doi.org/10.1145/3202185.3202741

Hilda, J. J., Jinny, V., & Saxena, M. (2023). Reinforcement Learning: Algorithms to balance the bias between Exploration and Exploitation. In 2023 Annual International Conference on Emerging Research Areas: International Conference on Intelligent Systems (AICERA/ICIS). 2023 Annual International Conference on Emerging Research Areas: International Conference on Intelligent Systems (AICERA/ICIS). IEEE. https://doi.org/10.1109/aicera/icis59538.2023.10420048

Ruiz, J., Gonzalez, C., Chen, Y., & Tang, B. (2023). Prize-Collecting Traveling Salesman Problem: A Reinforcement Learning Approach. In ICC 2023 - IEEE International Conference on Communications. ICC 2023 - IEEE International Conference on Communications. IEEE. https://doi.org/10.1109/icc45041.2023.10279682

Yeap, W. K. (2011). How Albot0 Finds Its Way Home: A Novel Approach to Cognitive Mapping Using Robots. In Topics in Cognitive Science (Vol. 3, Issue 4, pp. 707–721). Wiley. https://doi.org/10.1111/j.1756-8765.2011.01157.x

Zhu, H., Paschalidis, I. Ch., & Hasselmo, M. E. (2017). Feature extraction in Q-learning using neural networks. In 2017 IEEE 56th Annual Conference on Decision and Control (CDC). 2017 IEEE 56th Annual Conference on Decision and Control (CDC). IEEE. https://doi.org/10.1109/cdc.2017.8264148

Xu, C. (2023). Maze solving problem using q-learning. In Applied and Computational Engineering (Vol. 6, Issue 1, pp. 1491–1497). EWA Publishing. https://doi.org/10.54254/2755-2721/6/20230909