**Project 2**

Jasmine Villarreal

CS 370: Current/Emerging Trends

Professor Habibi

October 26th, 2025

**Describe the steps a human being would take to solve this maze.**

When it comes to problem-solving, the human mindset and machine learning take different approaches. A human mind would look at the maze and start visualizing patterns, testing routes, and recalling previous mistakes we might have made. This type of human testing relies on several factors, such as intuition, perception, and memory. An example would be a person tracing the maze and avoiding dead ends, and using logic to help predict where the goal is likely to be. (*Introduction to RL and Deep Q Networks*, 2023)

In contrast, an intelligent agent will not rely on intuition but rather learns from reinforcement learning. The agent will start without any prior knowledge, and with every episode completed, it will gradually improve its performance through repeated trial and error. By using Q-learning, the agent will update its Q-table or its neural network weight depending on the rewards the agent receives for each state-action pair. Over time, this process will enable the agent to help predict which actions are more favorable to lead to the treasure or main objective.

To solve the pathfinding task, the agent will need to interact repeatedly with the maze. For each step, the agent will evaluate according to the reward function, where reaching the goal will give a positive reward and invalid moves will incur penalties. This feedback will help the agent develop an algorithm through its experience, and this mirrors how humans use trial-and-error. As FreeCodeCamp(2018) explained, deep Q-learning allows an agent to improve by experiencing outcomes and making adjustments to its predictions, which mirrors how humans learn from repeated practice. The human mind may take a few attempts to solve the maze, but an agent likely needs hundreds of epochs to help generalize a solution through its data-driven learning.

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

The main objective of the agent was to find the shortest and most efficient route to the treasure, and to accomplish this, it would improve its decision-making over time. Reinforcement learning requires finding the right balance of exploration, in which the agent tries a couple of new actions, and exploitation, where it knows the best actions based on its past rewards (Contributor, 2023). In early training of the agent, it will explore frequently to gather more information about the maze. As it continues to train, exploration will decrease, and exploitation will increase to choose more suitable actions that will achieve the main objective.

In Codio, the exploration rate began high and slowly decayed as the model learned to reach the goal more efficiently. Training was set for 15,000 epochs, and the agent was able to achieve a strategy by 800 epochs successfully. This showed a strong convergence early on! Reinforcement learning used in this project depends on clear feedback to help shape the agent's learning. As the agent got closer to the treasure, it received a reward. When the agent moved farther away or into an obstacle, it would be penalized. This implemented feedback loop helped the agent to learn the most efficient path through repeated actions and interactions, and this is similar to how most people learn from trial and error.

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

The agent in this project utilized deep Q-learning, which is a combination of Q-learning and a neural network to help estimate future rewards for possible actions taken (*Reinforcement Learning (DQN) Tutorial — PyTorch Tutorials 2.7.0+Cu126 Documentation*, 2024). Traditional Q-learning will store every possible action pair in a table, but this approach is impractical in larger environments. Deep Q-learning replaces that table with a neural network that will help predict Q-values, allowing the agent to generalize across different states.

During the agent's training, its experiences are stored in a replay buffer, and this allows the model to sample past data in random batches for a stable learning approach (*Introduction to RL and Deep Q Networks*, 2023). A target network would update periodically to improve convergence and help prevent feedback instability. As the agent continued to learn, the model's loss decreased, and the win rate improved until the agent reached the treasure consistently.

The design for the reward system played a significant role as it showed that successful runs provided the agent with positive rewards. Invalid or failed runs resulted in the agent receiving a minor penalty, and this encouraged shorter paths. This reward system helps to motivate the agent to find the best route.

**Resources:**

*Introduction to RL and Deep Q Networks*. (2023). TensorFlow. https://www.tensorflow.org/agents/tutorials/0\_intro\_rl/

freeCodeCamp. (2018, April 11). *An introduction to Deep Q-Learning: let's play Doom*. FreeCodeCamp.org. https://www.freecodecamp.org/news/an-introduction-to-deep-q-learning-lets-play-doom-54d02d8017d8/

‌Contributor. (2023, October 10). *Reinforcement Learning: Balancing Exploration and Exploitation - insideAI News*. InsideAI News. https://insideainews.com/2023/10/10/reinforcement-learning-balancing-exploration-and-exploitation/

*Reinforcement Learning (DQN) Tutorial — PyTorch Tutorials 2.7.0+cu126 documentation*. (2024). Pytorch.org. https://docs.pytorch.org/tutorials/intermediate/reinforcement\_q\_learning.html