Project 2: Design Defense

Jermaine Wiggins

Southern New Hampshire University

CS-370: Current and Emerging Trends in Computer Science

Professor Divya Vellanki

Oct 19, 2025

**Project 2: Design Defense**

**Differences Between Human and Machine Approaches to Solving Problems**

Humans and AI approach maze-solving differently, although they share some similarities in how they learn. A human attempting to find the treasure in a maze would likely rely on trial and error, exploring various paths, marking dead ends, and gradually identifying the correct route. If the starting position changed, the human might need to explore new areas but could also use memory of previously successful routes to guide decisions. Similarly, the AI agent begins by exploring random paths but improves its decisions by updating Q-values, which represent the expected rewards for each action in each state (GeeksforGeeks, 2023). This process allows the AI to generalize its learning across every valid starting point in the maze. Both humans and AI learn through exploration and feedback, but the key difference lies in how knowledge is stored and used. The AI mathematically encodes all possible state-action pairs, ensuring consistent and optimal decisions, while humans depend on reasoning, memory, and intuition, which may vary based on maze complexity or starting positions (Milvus, n.d.).

**Purpose of the Intelligent Agent in Pathfinding**

In this treasure-hunting game, the intelligent agent, represented by a pirate, learns to navigate the maze through reinforcement learning. Exploration occurs when the pirate tries new actions to discover more efficient routes, while exploitation happens when the pirate uses learned knowledge to take the best-known action (Milvus, n.d.). Early in training, exploration is better as the pirate is unfamiliar with the environment, possible paths and outcomes. As learning progresses, exploitation is better because it allows the pirate to use what it has learned to efficiently follow the shortest and most reliable path to the treasure. This balance between exploration and exploitation is guided by the epsilon value, which starts high to encourage exploration and gradually decreases as the pirate learns more about the environment (GeeksforGeeks, 2023). Reinforcement learning supports this by giving the agent feedback/positive rewards for reaching the treasure and negative rewards for hitting walls or dead ends, which helps it determine the most effective strategy over time (Medium, 2023). Through this process, the pirate becomes increasingly efficient, ultimately mastering the maze and achieving near-perfect success rates.

**Implementing Deep Q-Learning**

Deep Q-learning was implemented using a neural network to approximate Q-values for each possible state-action pair. The pirate begins each training epoch at a random position and observes the current state of the maze. Actions are selected using an epsilon-greedy policy, which allows the pirate to balance exploration with exploitation (GeeksforGeeks, 2025). After performing each action, the resulting state, reward, and game status are stored in an experience replay memory. The neural network is then trained on batches of these experiences, which improves learning efficiency by reducing correlation between consecutive actions. A target network is used alongside the main model to stabilize training and prevent erratic updates (Medium, 2023).

As the pirate continues to play, epsilon decays gradually, shifting the agent’s behavior from exploration toward exploitation, meaning the pirate becomes more confident in following optimal routes. Over time, the model converges, and the pirate learns to find the treasure from any starting position with minimal steps (Milvus, n.d.). Tracking win rates ensures the training continues until performance reaches a consistent level, typically above 99 percent success. This implementation allows the agent to autonomously learn complex navigation behaviors that mirror adaptive human problem-solving while leveraging computational precision (GeeksforGeeks, 2023).

**References**

GeeksforGeeks. (2023). *Epsilon-Greedy Algorithm in Reinforcement Learning.* https://www.geeksforgeeks.org/machine-learning/epsilon-greedy-algorithm-in-reinforcement-learning/

GeeksforGeeks. (2025). *Deep Q-Learning in Reinforcement Learning.* https://www.geeksforgeeks.org/deep-learning/deep-q-learning/

Milvus. (n.d.). *What is an epsilon-greedy policy?* https://milvus.io/ai-quick-reference/what-is-an-epsilongreedy-policy

Milvus. (n.d.). *Exploration-exploitation tradeoff in reinforcement learning.* https://milvus.io/ai-quick-reference/what-is-the-explorationexploitation-tradeoff-in-reinforcement-learning

Medium. (2023). *Reinforcement Learning (DQN) Tutorial.* https://docs.pytorch.org/tutorials/intermediate/reinforcement\_q\_learning.html