**CS-370 Project Two: Design Defense**

Teisha Yoder  
Southern New Hampshire University  
 August 14, 2025

**Introduction:**

This project focused on creating an intelligent pirate agent using deep Q learning to navigate an 8×8 maze environment and find a treasure located at a fixed position. The maze was defined as a NumPy 2D array marking free and blocked cells and visualized with a custom function that distinctly displayed visited locations, the pirate's current position, and the treasure cell. The agent could move in four directions (left, up, right, and down). It began training with an exploration factor (epsilon) of 0.1, allowing random exploratory actions approximately 10% of the time during training.

The main aim was to successfully train the agent to autonomously discover optimal paths based on rewards, reliably reaching the treasure from any starting point. This aligns with reinforcement learning principles, where agents learn optimal behaviors through interaction with their environment (Sutton & Barto, 2018).

**Human vs. Machine Problem Solving Approaches:**

Human Approach  
A typical human maze solver would:

* Visually assess open routes versus obstacles.
* Use prior experience and spatial reasoning to select promising paths.
* Maintain mental awareness of visited areas to avoid loops or backtracking unnecessarily.
* Make logical decisions to backtrack when encountering dead ends.
* Continue until reaching the treasure efficiently.

Humans utilize cognitive planning and intuition to solve such problems with minimal trial-and-error.

Agent Approach  
Conversely, the agent starts with no maze knowledge and applies deep Q learning (Mnih et al., 2015) to:

* Encode the current state of the maze.
* Choose actions with an epsilon-greedy strategy (exploring randomly or exploiting the learned policy).
* Receive immediate rewards or penalties reflective of progress towards the goal.
* Store observational episodes in experience replay memory.
* Train a neural network Q function approximator with batches of experiences to update value predictions.
* Iterate over many episodes, refining the policy toward optimal navigation strategies.

Similarities and Differences  
Both methods rely on iterative learning from experience. Humans bring intuition and heuristic shortcuts, enabling rapid strategic planning, while the agent systematically improves through numerous simulations, exploring exhaustively without intuition but with computational thoroughness.

**Purpose of the Intelligent Agent in Pathfinding:**

The agent’s purpose is to fully autonomously learn efficient navigation policies in the maze that generalize across start positions without manual path coding. This adaptability enhances game dynamics by providing an intelligent, competitive NPC that functions robustly in spatial navigation tasks shaped by reinforcement signals (Neacsu & Franco, 2024).

**Exploitation vs. Exploration in Pathfinding:**

* Exploration involves trying actions that might be less familiar or risky to discover better paths.
* Exploitation focuses on leveraging known high value actions to maximize reward.

Training began with epsilon at 0.1, facilitating moderate exploration of unknown routes. Upon surpassing a 90% win rate threshold, epsilonreduced to 0.05, strengthening exploitation of learned effective paths. Maintaining this careful balance prevented the agent from prematurely converging on suboptimal strategies while encouraging comprehensive learning early on (Sutton & Barto, 2018).

**Reinforcement Learning and Pathfinding:**

Reinforcement learning framed pathfinding as a value maximization problem:

* The agent earned small negative step penalties to encourage efficiency, with positive rewards for reaching the treasure. Invalid moves resulted in penalties to discourage inefficient actions.
* Experience replay sampled past episodes for training, reducing correlation bias and improving policy generalization (Mnih et al., 2015).
* The Bellman equation iteratively updated Q values as expected rewards for actions considering both immediate outcomes and future gains.

This approach enabled the trained agent to generalize an optimal policy valid from any free starting position, which was confirmed by a comprehensive completion check testing all valid positions.

**Deep Q-Learning Implementation and Results:**

Model Architecture:

* Input Layer: 64 neurons matching the maze cells in the 8×8 grid.
* Hidden Layers: Two Dense layers of 64 neurons each with Parametric ReLU activation.
* Output Layer: 4 neurons representing possible moves.
* Loss Function: Mean Squared Error.
* Optimizer: Adam.

**Training Setup:**

* Maximum Epochs: 15,000.
* Experience Replay Memory Size: 1,000 episodes.
* Batch Size: 24; training epochs per batch: 8.
* Data Size: 50.
* Epsilon: 0.1 initially, dropping to 0.05 when win rate exceeded 90%.

Performance Summary:

* Initial epochs exhibited low win rates due to extensive exploration.
* The agent progressively improved through many episodes, optimizing moves based on received feedback.
* After crossing 90% win rate, exploration was decreased to prioritize exploiting known good policies.
* Completion checks verified 100% success in reaching the treasure from all valid start locations.
* Loss values decreased consistently, indicating learning stability.
* Training took several minutes, with logged outputs showing epochs, loss, episode counts, win totals, win rate, and elapsed training time.

**Conclusion:**

This project demonstrated that deep Q learning successfully addresses complex spatial pathfinding challenges by enabling an agent to learn effective navigation policies solely through reward based experience. The balance of exploration and exploitation, supported by experience replay and neural net function approximation, facilitated robust generalization and convergence to a perfect success rate. The trained pirate agent’s ability to consistently find the treasure from any starting point illustrates reinforcement learning’s strength in creating adaptive and intelligent game NPC behaviors.

**References:**

* Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature, 518*(7540), 529–533. <https://doi.org/10.1038/nature14236>
* Neacsu, A., & Franco, S. (2024). Reinforcement learning for intelligent pathfinding in dynamic environments. *International Research Journal of Modernization in Engineering Technology and Science, 6*(11), 4128–4136. <https://www.irjmets.com>
* Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*(2nd ed.). MIT Press. <https://incompleteideas.net/book/the-book-2nd.html>