# **Project Two: Design Defense**

Mark Baligad

Southern New Hampshire University

CS-370 Current/Emerging Trends in CS

Professor Joe Oakes

10/19/2025

## Design Defense

### I. Introduction

In this project, we created an Intelligent Pirate Agent that navigates an 8x8 maze to reach a target treasure cell. The project uses a machine learning Python library in conjunction with a Q-Learning algorithm to train the agent to find the correct path. Moving out of bounds of the maze, running into a wall, backtracking, and visiting already explored cells are each assigned penalties to encourage valid movement. For my solution, I replaced the Keras module with PyTorch to take advantage of its support for the latest CUDA 12.8 toolkit with Blackwell GPU support. This change drastically reduced training time from several hours to just minutes on my system while still meeting the requirement to train the agent using a Q-learning algorithm and neural network.

### II. Differences in Human and Machine Approaches

A human, assuming they have no prior knowledge of the maze layout, would approach the maze by identifying dead ends and planning new routes. A human might start by following a straight path until they meet an obstacle, then backtrack to the last open path and try another direction. Over time, a human builds a memory of failed paths and avoids repeating the same mistakes. This process relies on reasoning, observation, and memory rather than strict mathematical optimization.

A machine agent takes a different approach to solving the same problem. The agent must attempt many possible moves without understanding which ones are good or bad until it gathers enough experience. It relies on trial and error to associate actions with rewards or penalties. Early on, the agent moves almost randomly, but over many training episodes, it begins to favor moves that lead to higher rewards.

While both humans and machines use feedback from mistakes to improve, the difference lies in how that learning occurs. Kumar et al (2022) explain that humans generally perform better at abstract tasks and worse at pattern-matching, while a neural network excels at pattern-matching and struggles with abstract tasks. Humans use logic and memory to adjust behavior quickly, while the machine must calculate and remember numerical reward values for each possible action. A human could often solve the maze on the first attempt using their abstract knowledge of pathfinding. At the same time, the agent may need hundreds or thousands of attempts to find an efficient route based on identifying patterns across each cell of the maze.

### III. Purpose of the Intelligent Agent in Pathfinding

The purpose of the intelligent agent is to learn an effective strategy for navigating from the starting point to the treasure cell without any prior map or instructions. Through reinforcement learning, the agent balances exploration (trying new moves) and exploitation (reusing what it has learned). In the beginning, it explores freely, trying many paths and receiving penalties for bad actions and rewards for progress. As learning continues, the agent begins to exploit its experience to choose moves that are more likely to succeed.

Reinforcement learning helps the agent determine which actions are valuable based on their outcomes. As Serengeti Software Technologies (2019) explains, reinforcement learning focuses on finding the best action in a given situation through trial and feedback rather than pre-labeled examples. The agent must act, observe, and learn from the resulting rewards or penalties. The agent updates a table of Q-values that represent how good each possible action is from a given position. Over many training rounds, these Q-values become more accurate, guiding the agent toward the goal more efficiently. The result is a model that can navigate the maze successfully with minimal random behavior once trained.

### IV. Use of Algorithms to solve complex problems

To train the intelligent agent, a Deep Q-Learning (DQL) algorithm combined with a neural network was implemented. This approach allows the agent to learn how to navigate the maze through trial and error rather than explicit instructions. The neural network predicts Q-values for each possible move based on the agent’s current state, estimating which action will lead to the greatest future reward. During training, these predictions are continuously refined using backpropagation to minimize the gap between expected and actual outcomes. Over many episodes, the pirate agent gradually learns which paths lead closer to the treasure and which actions to avoid, transforming random exploration into purposeful decision-making.

This algorithm demonstrates how reinforcement learning can solve complex problems that lack predefined solutions. By interacting with the environment, receiving feedback, and updating its strategy, the agent develops goal-directed behavior similar to human learning. As Serengeti Tech (2019) explains, “The purpose of Q-learning algorithm is to learn a quality function that will give us correct action for a given state.” This principle applies beyond games; the same learning process underlies real-world applications such as robotic navigation, autonomous driving, and warehouse route optimization. Through this project, the Deep Q-Learning model shows how intelligent systems can adapt to changing environments and make effective decisions based purely on experience.

## References

Kumar, M., Zaharia, M., & Yamins, D. L. K. (2022). *Disentangling abstraction from statistical pattern matching in human and machine learning*. arXiv preprint arXiv:2204.01437. <https://arxiv.org/abs/2204.01437>

Serengeti. (2019). Using Q-Learning for pathfinding. *Serengeti*. <https://serengetitech.com/tech/using-q-learning-for-pathfinding/>