**CS 370 Project Two Design Defense: Pirate Intelligent Agent**

**Ifeoluwa Adewoyin**  
**CS 370**  
**Project Two - Treasure Hunt Game**  
**08/23/25**

**Introduction**

This design defense examines the development and implementation of an intelligent pirate agent for a treasure hunt game using deep Q-learning. The project demonstrates fundamental artificial intelligence concepts including reinforcement learning, neural networks, and pathfinding algorithms. The intelligent agent successfully learns to navigate an 8x8 maze environment to reach a treasure while avoiding obstacles and maximizing rewards through experience-based learning.

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

**Human Problem-Solving Approach**

When faced with solving the treasure hunt maze, a human player would typically employ several cognitive strategies. First, humans would visually scan the entire maze to identify the starting position, treasure location, and obstacles. They would mentally trace potential paths from start to finish, likely identifying the most direct route while noting dead ends and blocked passages.

Human problem-solving in this context involves spatial reasoning, pattern recognition, and strategic planning. A human would likely use a combination of forward planning (planning moves from current position toward the goal) and backward chaining (working backward from the treasure to find viable paths). Humans also benefit from their ability to see the "big picture" immediately, allowing them to develop a general strategy before executing specific moves.

Additionally, humans would learn from mistakes through trial and error, but with the advantage of conscious reflection. They can analyze failed attempts, identify what went wrong, and deliberately adjust their strategy for subsequent attempts.

**Machine Learning Approach**

The intelligent pirate agent approaches the same problem through deep Q-learning, a form of reinforcement learning. Unlike humans, the agent begins with no prior knowledge of the maze structure or optimal pathfinding strategies. Instead, it learns through interaction with the environment, receiving feedback in the form of rewards and penalties.

The agent's learning process involves exploration (randomly trying different actions to discover new paths) and exploitation (using learned knowledge to make optimal decisions). The deep neural network serves as the agent's "brain," learning to map environmental states to action values (Q-values) that represent the expected future reward for each possible action.

The machine learning approach is fundamentally iterative and statistical. The agent performs thousands of episodes, gradually building a comprehensive understanding of state-action relationships through experience replay and neural network training.

**Similarities and Differences**

Both approaches involve learning from experience and adapting strategies based on outcomes. Both must balance exploration of new possibilities with exploitation of known successful strategies. Additionally, both approaches ultimately seek to minimize the number of steps required to reach the treasure while avoiding obstacles.

However, the key differences lie in their learning mechanisms and initial knowledge. Humans benefit from spatial reasoning, immediate pattern recognition, and the ability to form abstract strategies. The machine learning approach requires extensive training data and computational iterations but can process complex state-action relationships that might overwhelm human cognitive capacity. Humans learn more efficiently with fewer examples but are limited by cognitive biases and processing capacity. The AI agent requires many more training episodes but can achieve consistent, optimal performance once trained.

**Purpose of the Intelligent Agent in Pathfinding**

**Exploration vs. Exploitation**

Exploration refers to the agent's tendency to try new, potentially suboptimal actions to discover better strategies and gain information about the environment. In the treasure hunt game, exploration means the pirate might choose a random direction even when it has learned that another direction typically yields better results. This randomness is controlled by the epsilon parameter in the epsilon-greedy strategy.

Exploitation, conversely, involves using current knowledge to make the best possible decisions based on learned experience. When exploiting, the pirate chooses actions with the highest predicted Q-values, following the path it believes will lead to the treasure most efficiently.

**Ideal Exploration-Exploitation Balance**

For this pathfinding problem, the implemented epsilon-greedy strategy provides an effective balance. The initial epsilon value of 0.1 means the agent explores randomly 10% of the time and exploits learned knowledge 90% of the time. This proportion works well because the maze environment is relatively small (8x8) and deterministic.

The adaptive epsilon reduction to 0.05 when the win rate exceeds 90% demonstrates sophisticated balance optimization. Early in training, more exploration helps the agent discover viable paths and avoid local optima. As performance improves, reduced exploration prevents the agent from abandoning successful strategies while still allowing occasional discovery of potentially better alternatives.

This balance is ideal because it ensures thorough exploration of the state space while progressively focusing on optimal policies. Too much exploration would prevent convergence to optimal behavior, while too little would risk missing superior strategies or failing to adapt to environmental changes.

**Reinforcement Learning in Pathfinding**

Reinforcement learning enables the pirate agent to determine optimal paths through the reward feedback mechanism. The agent receives positive rewards (+1.0) for reaching the treasure, negative penalties for hitting obstacles (-0.75), attempting to move outside boundaries (-0.8), or revisiting cells (-0.04). These rewards guide the learning process by reinforcing successful behaviors and discouraging counterproductive actions.

The Q-learning algorithm updates the agent's understanding of state-action values based on the Bellman equation, incorporating both immediate rewards and estimated future values. This temporal difference learning allows the agent to develop policies that consider long-term consequences, not just immediate rewards.

Through experience replay, the agent can learn from past experiences multiple times, improving sample efficiency and stabilizing learning. The neural network generalizes across similar states, enabling the agent to make intelligent decisions in previously unencountered situations while building toward optimal pathfinding strategies.

**Implementation of Deep Q-Learning Using Neural Networks**

**Neural Network Architecture**

The deep Q-learning implementation utilizes a feed-forward neural network with three layers. The input layer accepts environmental states with dimensionality equal to the maze size (64 neurons for the 8x8 maze). Two hidden layers, each with 64 neurons, use Parametric Rectified Linear Unit (PReLU) activation functions. The output layer contains four neurons corresponding to the four possible actions: left, right, up, and down.

This architecture provides sufficient capacity to learn complex state-action mappings while remaining computationally efficient. The PReLU activation functions help mitigate the vanishing gradient problem and allow the network to learn negative activations when beneficial.

**Training Process and Experience Replay**

The implementation incorporates experience replay, a crucial component for stable deep Q-learning. The GameExperience class stores episodes consisting of state transitions, actions, rewards, and terminal conditions in a replay buffer with maximum capacity of 512 experiences (8 × maze size).

During each training iteration, the agent samples 32 experiences from the replay buffer to train the neural network. This approach provides several advantages: it breaks correlation between consecutive experiences, enables multiple learning updates from single experiences, and stabilizes training by providing diverse training samples.

**Q-Learning Algorithm Implementation**

The Q-training algorithm follows the standard deep Q-learning approach with several key optimizations. For each epoch, the agent randomly selects a starting position, ensuring comprehensive exploration of the state space. The epsilon-greedy policy balances exploration and exploitation, with epsilon adaptively decreasing as performance improves.

The neural network uses the Adam optimizer with mean squared error loss function, providing efficient gradient-based learning. The target Q-values are calculated using the Bellman equation, incorporating both immediate rewards and discounted future value estimates.

The training process continues until the agent achieves 100% win rate across all possible starting positions, verified through the completion check function. This ensures the learned policy is robust and generalizes across the entire maze environment.

**Conclusion**

The successful implementation of the pirate intelligent agent demonstrates the effectiveness of deep Q-learning for pathfinding problems. The agent achieved optimal performance through the systematic application of reinforcement learning principles, neural network function approximation, and experience replay mechanisms.

The comparison between human and machine approaches reveals complementary strengths: human intuition and spatial reasoning versus machine learning's systematic exploration and optimization capabilities. The careful balance of exploration and exploitation, implemented through adaptive epsilon-greedy strategies, proved crucial for achieving optimal performance while maintaining learning efficiency.

This project illustrates how artificial intelligence can solve complex navigation problems through iterative learning and neural network approximation, providing insights applicable to broader domains including robotics, autonomous systems, and game AI development.

**References**

Russell, S., & Norvig, P. (2020). *Artificial intelligence: A modern approach* (4th ed.). Pearson.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction* (2nd ed.). MIT Press.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529-533.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.