**Design Defense: Pirate Intelligent Agent for Pathfinding Problem**

The pirate intelligent agent in the treasure hunt game is tasked with navigating a maze to find the treasure before the player. The agent is trained using a reinforcement learning technique called deep Q-learning, which allows it to improve its pathfinding abilities through experience. This paper explains the approach used to solve the pathfinding problem, comparing human and agent problem-solving methods, and discusses the role of exploration and exploitation in the agent’s learning. Lastly, I will evaluate the deep Q-learning algorithm implemented for this game.

**Human vs. Machine Approach to Solving the Maze**

Human Steps to Solve the Maze A human solving the maze would observe the layout, identify obstacles, and plan a route to the treasure. They may use memory and reasoning to avoid revisiting dead ends and adapt their strategy based on past decisions. Humans can rely on foresight, breaking the maze down into smaller, manageable sections to gradually reach the goal.

Pirate Intelligent Agent’s Steps In contrast, the pirate agent relies on trial and error through deep Q-learning. It starts without knowledge of the environment and learns by interacting with the maze over multiple episodes. It receives rewards for making progress and penalties for hitting walls or illegal moves. Through exploration (trying new paths) and exploitation (choosing the best-known path), the agent gradually learns to optimize its actions to reach the treasure.

Similarities and Differences Both the human and the agent aim to reach the treasure efficiently, but humans leverage cognitive skills like reasoning and foresight, while the agent depends on repeated experience and reward-based feedback. The agent requires many trials to improve, unlike a human who can solve the maze with fewer attempts by planning ahead.

**Purpose of the Intelligent Agent in Pathfinding**

Exploitation vs. Exploration In reinforcement learning, exploitation refers to the agent choosing actions based on learned experiences, while exploration involves trying new actions to discover better paths. The epsilon parameter controls this balance. In this project, epsilon was set to 0.1, meaning the agent explored 10% of the time and exploited 90% of the time. This balance allowed the agent to avoid becoming stuck in suboptimal paths while maximizing the chances of finding the treasure over time.

Reinforcement Learning and Pathfinding The pirate agent uses reinforcement learning to improve its pathfinding abilities. It learns by receiving rewards for successful actions and penalties for errors. The Q-learning algorithm updates the agent’s policy by assigning higher values to actions that lead to positive outcomes. Over time, the agent refines its policy to choose actions that consistently bring it closer to the treasure.

**Use of Algorithms to Solve Complex Problems**

Deep Q-learning Implementation Deep Q-learning approximates the expected reward (Q-value) for each action using a neural network. The network takes the maze’s current state as input and outputs the Q-values for the four possible actions (left, right, up, and down). The agent selects the action with the highest Q-value or explores a random action based on epsilon. As the agent navigates the maze, it stores experiences (state, action, reward, next state) in memory, which are used to update the network and improve future decision-making.

The agent’s performance improved after training on thousands of episodes, as it learned to associate certain actions with higher rewards. By the end of training, the agent could consistently navigate the maze to the treasure, demonstrating how deep Q-learning efficiently solves complex problems like pathfinding.

**Conclusion**

The pirate intelligent agent successfully learned to navigate the maze and find the treasure using deep Q-learning. The balance between exploration and exploitation enabled the agent to learn from both trial and error and optimal decision-making. Through reinforcement learning, the agent improved its performance over time, highlighting the effectiveness of AI algorithms in solving complex pathfinding problems.

**References**

Yoon, C. (2018a). Deriving Policy Gradients and Implementing REINFORCE. Medium. <https://medium.com/@chrisyoon7/deriving-policy-gradients-and-implementing-reinforce-519b49d13c9e>

Yoon, C. (2018b). Understanding Actor Critic Methods and A2C. Towards Data Science. <https://towardsdatascience.com/understanding-actor-critic-methods-and-a2c-using-detailed-implementations-9bc751eb5b4e>

Surma, G. (2020). Cartpole - Introduction to Reinforcement Learning (DQN - Deep Q-Learning). Medium. <https://medium.com/@gsurma/cartpole-introduction-to-reinforcement-learning-dqn-deep-q-learning-779f8a087feb>