**Design Defense Document**

**Introduction**

The objective of this project is to develop an intelligent agent for a pirate non-player character (NPC) in a treasure hunt game. The agent navigates a maze environment to locate a treasure while competing against a human player. This document explains the approach to solving the problem, highlights the algorithm used, and evaluates its performance.

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

**Steps a Human Would Take**

A human solving the maze would:

1. Visualize the maze layout and identify obstacles and paths.

2. Use trial and error to explore pathways, marking visited locations mentally or physically.

3. Prioritize unvisited paths and move towards the goal based on intuition.

**Steps Taken by the Intelligent Agent**

The pirate agent employs reinforcement learning to solve the problem:

1. Observes the maze as a 2D array and the current state as the agent’s position.

2. Uses a deep Q-learning algorithm to learn the best actions through trial and error.

3. Balances exploration (trying new paths) and exploitation (using known paths).

4. Receives rewards based on its actions and adjusts its strategy accordingly.

**Similarities and Differences**

- **Similarities:** Both approaches involve exploration of pathways and avoiding previously visited dead ends.

- **Differences:** The human relies on intuition and visual cues, while the agent uses mathematical models and learns through iterative trials.

**Purpose of the Intelligent Agent in Pathfinding**

**Exploitation vs. Exploration**

- **Exploitation:** Refers to using the learned knowledge to maximize rewards by taking the best-known actions.

- **Exploration:** Involves trying new or random actions to discover potentially better strategies.

- **Ideal Proportion:** For this problem, an exploration factor (ε) of 0.1 is appropriate. Early in training, more exploration is beneficial, but over time, the agent should exploit learned paths for efficiency.

**Role of Reinforcement Learning**

Reinforcement learning helps the agent:

- Assign value to each action based on rewards received.

- Use these values to create a policy for selecting the best action in any given state.

- Optimize its pathfinding through iterative updates to its neural network.

**Deep Q-Learning Implementation**

The deep Q-learning approach involves:

**1. Model Architecture:** A neural network with two hidden layers is used to approximate Q-values for state-action pairs.

**2.** **Game Experience:** Stores and samples past episodes to train the model efficiently, ensuring the agent learns from varied scenarios.

**3. Training Loop:**

- The agent interacts with the maze and stores experiences in memory.

- It uses these experiences to train the network, adjusting weights to minimize loss between predicted and target Q-values.

**Conclusion**

The pirate intelligent agent successfully learns to navigate the maze using deep Q-learning. By balancing exploration and exploitation and iteratively improving its strategy, the agent achieves its goal of finding the treasure efficiently. The use of reinforcement learning demonstrates the power of AI in solving complex pathfinding problems.