**Project Two Design Defense**

**Justin Hancock**

**CS-370 Current & Emerging Trends**

**06.18.2024**

Throughout the development of the intelligent agent in this treasure hunt game, it is important to understand the fundamental AI concepts that supports the agent’s ability to navigate the game world effectively. My design defense explores the distinctions between human and machine problem solving approaches that describe the steps both a human and the agent take to solve the maze, compare the two approaches, assess the purpose and mechanisms of the agent, and evaluate the role of reinforcement learning. I will conclude by diving into the implementation details of deep Q-learning using neural networks.

Humans solve problems using intuition, experience, and pattern recognition. We initially rely on visual cues to choose paths, learn from trial and error, and adjust strategies based on new information. For example, we would learn by playing the game and finding where the dead ends are located (trial and error). We would then use memory from trial and error to avoid the dead end and find a new path to the end of the maze. Machines, on the other hand, follow algorithmic rules that leave out the intuition and focus on a structured approach to memory and reward-based learning. While the machines approach still uses memory, the systematic exploration uses algorithms to attempt all possible pathways. The comparison of the two approaches highlights the machine’s consistency and structured learning versus our flexibility and adaptability.

A human solving the maze would start with a visual scan to identify a potential path with obstacles that would be in the way. We would choose an initial path based on our experience to proceed and backtrack when encountering a dead end. Throughout the game, we would remember the unsuccessful paths to avoid making the same mistake and recognize patterns to predict and elude dead ends. Our ability to adapt is what enables us to refine our strategy continuously.

The intelligent agent begins by initializing its state and Q-values. It balances exploration and exploitation by selecting actions randomly to explore the environment and then choosing the best-known actions based on its learned Q-values. After each action, the agent transitions into a new state, collects the rewards or penalties, and updates the Q-values accordingly. The process allows the agent to shift from exploration to exploitation by improving its pathfinding strategy over time.

The primary purpose of the intelligent agent in pathfinding is to autonomously navigate the game world and find the treasure before the human player. It aims to learn the optimal path through reinforcement learning, adapt to various maze configurations, and apply learned strategies to maximize efficiency towards reaching the goal. It enables the agent to perform at or above human levels in specific pathfinding tasks. Our study last week on AlphaGo Zero is a prime example of the purpose of intelligent agent pathfinding. The concept can directly be applied to our treasure hunt game.

Exploitation uses known information to choose the best action to maximize the reward. Exploration tries new techniques and actions to discover better paths and rewards. Ideally, you would want your agent to have high exploration in early learning to discover the environment, and more exploitation in later training to refine the learned path. By using this training strategy, exploration allows the agent to gather the proper information about the environment, and exploitation applies the learned information to consistently choose the best paths towards the goal.

Reinforcement learning aids the agent in determining the path to the goal by assigning rewards for reaching the treasure and penalties for dead ends or obstacles. The agent continuously updates the Q-values to reflect the state/action combination and learns from experiences to improve its strategy over time. Reinforcement learning allows the agent to discover and optimize paths that guide it to the treasure.

Algorithms like Deep Q-learning are effective in solving complex problems due to their ability to systematically explore and learn from large state spaces. These types of algorithms often achieve and outperform humans in specified tasks by learning optimal strategies through repeated trials. The implementation of deep Q-learning using neural networks involves several steps. Parameters such as learning rate, discount factor, and exploration are first initialized. A neural network model is then built with layers to approximate the Q-values. An experience replay will store memory of past experiences for training. During the training process, the agent transitions between states, collects rewards, and updates the Q-values. The neural network is trained using these experiences. It also shifts from exploration to exploitation to refine the pathfinding strategy. When the approach is structured like this, the agent can learn efficiently and effectively to navigate the game.

**Citations**

Colyer, A. (2016, September 20). *Mastering the game of Go with deep neural networks and tree search | the morning paper*. https://blog.acolyer.org/2016/09/20/mastering-the-game-of-go-with-deep-neural-networks-and-tree-search/

Hk, A. (2024, March 13). Mastering Deep Q-Learning with PyTorch: A Comprehensive Guide. *Medium*. https://medium.com/@hkabhi916/mastering-deep-q-learning-with-pytorch-a-comprehensive-guide-a7e690d644fc

Yax, N., Anlló, H., & Palminteri, S. (2024). Studying and improving reasoning in humans and machines. *Communications Psychology*, *2*(1). https://doi.org/10.1038/s44271-024-00091-8