Design Defense: Pirate Intelligent Agent

Noah Khomer

Southern New Hampshire University

CS-370- Current and Emerging Trends in CS

Timothy Alexander

7/20/2025

Design Defense: Pirate Intelligent Agent

There is a significant difference in human and machine problem-solving. The reason for this is because they are approached to solve problem-solving itself very differently. For example, a human is going to navigate through the maze by relying on vision, spatial memory, and even the scene-making. They will look around, trace back, and apply the logic to find the shortest path based on their instinct.

On the other hand, a machine is quite different, especially an intelligent agent. For example, the pirate that I used for my final project will try to solve problems by using data and algorithms that can learn from experience. For my project, the pirate agent uses a deep Q-learning algorithm to find where the treasure is in a maze. At the start, it really starts with no knowledge and learns by trial and error but tries to observe the states, for example, the positions in the maze, and then takes action based on that, like moving up, moving down, going left, or going right.

The agent also receives rewards, such as a positive number for finding the treasure and a negative number for any invalid move or step that it makes, and then finally approaches updating the Q value using a neural network. As time goes on, this process forms an understanding of which paths are most likely to lead to success.

So, what is the purpose of intelligent agents and pathfinding? The pirate agent's role is to solve a pathfinding problem by learning the most effective way so that it can reach the treasure before the player, which is a human, can. The reason it can do this is because of reinforcement learning, which is going to train the agent on how to maximize long-term reward based on the actions in each environment.

It can also be learned from sequences of states’ actions and outcomes. One of the key concepts in reinforcement learning is that it can go between exploration and exploitation. In exploration, the agent is always trying new paths to learn more about the data, and then exploitation is always choosing the best-known actions it has already tried. Reinforcement learning can help the agent determine which part is going to get to the treasure by repeating the training episodes that are then stored as experiences.

These experiences can be anything from state action rewards to even the next date so that they can associate better action with higher future rewards. As it tries to get higher future rewards, it will be able to find the treasure more efficiently over time.

For my project I have used the deep Q-learning algorithm so that I can add a neural network to estimate the Q values. The reason for this is a Q-table becomes more impractical for any type of large state space.

Therefore, within your network, the agent can use the model with dense layers and activation functions so that it can approximate the best action for the given state. Then, in the training, the agent can feed the state information into the network. The network then tries to predict the value of each action. Once the network can take an action and observe the result, it updates the prediction model by using the Bellman equation and backpropagation.

As the model becomes accurate enough to guide the agent through the maze reliably, model also includes a layer of PReLU activation, and it can optimize speed for convergence. The training can run over hundreds and even thousands of episodes to ensure the agent is able to explore the maze enough so that it can adjust to the situations inside it. This approach is very scalable because it gives us a strong solution for a complex environment that the agent is always going to experience in a maze-like scenario.

References

Amin, S. (2024, September 14). *Deep Q-Learning (DQN).* *Medium.* <https://medium.com/@samina.amin/deep-q-learning-dqn-71c109586bae>

*The deep Q-learning algorithm - hugging face deep RL course.* (n.d.). *Hugging Face.* <https://huggingface.co/learn/deep-rl-course/en/unit3/deep-q-algorithm>

GeeksforGeeks. (2024, May 18). *Exploitation and exploration in machine learning.* <https://www.geeksforgeeks.org/exploitation-and-exploration-in-machine-learning/>

Harvard Business School. (2023, November 22). *Humans vs. machines: Untangling the tasks AI can (and can’t) handle: Working knowledge.* *Harvard Business School.* <https://www.library.hbs.edu/working-knowledge/humans-vs-machines-untangling-the-tasks-ai-can-and-cant-handle>

Karunakaran, D. (2020, September 21). *Deep Q network (DQN) - applying neural network as a functional approximation in Q-learning.* *Medium.* <https://medium.com/intro-to-artificial-intelligence/deep-q-network-dqn-applying-neural-network-as-a-functional-approximation-in-q-learning-0ffe3b0a9062>

*Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction* (2nd ed.). *Stanford University.* <https://web.stanford.edu/class/psych209/Readings/SuttonBartoIPRLBook2ndEd.pdf>

Amazon Web Services. (n.d.-b). *What is reinforcement learning? – Reinforcement learning explained.* *AWS.* <https://aws.amazon.com/what-is/reinforcement-learning/>