Ty Wheeler

4-2 Milestone Three Narrative

CS 499

5/26/25

*Artifact description*

For this artifact I selected the Pirate Intelligent Agent project that I developed in my CS 370 course. This project simulates a reinforcement learning agent that navigates a maze to find treasure using Q-learning. The agent originally used a standard Deep Q-Network implementation trained on environment feedback, and storing past episodes for experience replay. The maze environment, agent behavior and training logic were implemented in Python and executed in a Jupyter Notebook. This project was created earlier in my program but has now been enhanced with more advanced techniques as well as data visualization.

*Justification for inclusion in ePortfolio*

I chose this artifact for my ePortfolio because it demonstrates my ability to apply advanced reinforcement learning methods and optimize algorithm performance. These are skills that are central to the learning outcomes for algorithms and data structures. I enhanced the project by implementing Double Deep Q-Networks, Double DQN, to reduce overestimation bias and improve the stability of training overall. I also added real-time visualization tools to better interpret the agent’s learning process, including a cell-based heatmap of Q-values and a line graph showing average Q-values per episode. These enhancements showcase my ability to critically evaluate algorithm behaviors, improve learning performance through architectural changes and use visualization to analyze large sets of data at a glance.

The enhancements reflect the application of algorithmic thinking and data structure analysis. For example the transition to Double DQN required changes to how experience replay data is processed, leveraging both a current and target neural network to decouple action selection from the evaluation. This artifact aligns with course outcomes related to the application of algorithmic principles, the use of appropriate data structures, and the creation of intelligent systems using industry techniques.It demonstrates my ability to critically assess algorithmic performance by identifying and addressing overestimation in Q-value updates through the implementation of Double DQN. It also reflects my application of advanced reinforcement learning techniques, such as the decoupling of action selection from evaluation to improve training stability.

*Reflection on the enhancement process*

Enhancing this agent taught me a lot about reinforcement learning stability and algorithm behavior. Implementing Double DQN required modifying the GameExperience class to incorporate a separate target network. I adjusted the get\_data() method to select actions using the main model and evaluate Q-values with the target model. I also introduced periodic updates to sync the target network with the current model, which involved tracking training steps and maintaining consistency across weight copies. For the visualization enhancement, I created a “show\_q\_values()” function using Matplotlib that displays the best Q-value per cell directly on the maze grid. This helped make the agent's learning more interpretable for users at a glance. I also tracked the average predicted Q-value for each episode during training and plotted these values after training finished. This graph acts as a learning curve showing trends in the agent’s value predictions over time. One of the main challenges I encountered was verifying that the Double DQN behavior was actually implemented correctly. Since changes were internal to the learning algorithm, debugging relied on close monitoring of training logs and Q-value trends. Another challenge was integrating visualization logic without disrupting the existing training output. I overcame this by isolating the visualizations to separate code blocks and ensuring qtrain() returned relevant statistics for plotting. Overall this project reinforced my understanding of reinforcement learning principles, introduced more stable Q-value update techniques, and improved my ability to communicate AI learning behavior through visuals. It demonstrates not only how I can implement technically sound improvements but also how I can critically analyze and interpret the behavior of intelligent systems.