Randy Ortiz

Professor Kraya

CS-499 Computer Science Capstone 2025

July 31, 2025

The artifact selected for this category is a reinforcement learning project developed in Jupyter Notebook using Python and TensorFlow/Keras. The program simulates a pirate agent learning to navigate an 8x8 maze to reach a treasure using a Deep Q-Learning algorithm. Upon launch, the user defines the maze and builds a model using a neural network, which the agent trains over several epochs. Each training episode involves the agent selecting actions (up, down, left, right) and receiving feedback in the form of rewards or penalties based on its path. The core logic is handled through a combination of environment simulation (TreasureMaze.py), replay memory (GameExperience.py), and training through the qtrain() function.

The enhancement I made to this artifact was to integrate a MongoDB database to log the training metrics after each epoch. In the original artifact, the models’ performance data (win rate, number of episodes, losses, win history) was only printed to the console which would then be lost when the kernel is reset. To enhance it, I created a MongoDB database named RLTrainingDB and a collection called training\_metrics. Within the training loop, I inserted a code block that logs the metrics as documents in the collection. I also implemented connection checks and printed the available databases and collections to confirm successful integration and log visibility.

This enhancement aligns with Course Outcome 2 and Course Outcome 4. For Outcome 2, it shows my ability to design and implement technical enhancements with clear communications through structured logging and database integration. For Outcome 4, the use of pymongo to store training data in a MongoDB collection so it can be retrieved and analyzed later shows my proficiency in applying real-world tools to support computing solutions.

The process also strengthened my skills in debugging and working with external tools. One challenge I faced was a datetime naming conflict when trying to log timestamps. The error happened because I imported datetime two different ways, which caused confusion in the code. I fixed it by cleaning up the imports and using datetime.now() correctly. Another issue I encountered involved outdated keras imports. The original code used from keras.models import Sequential and from keras.layers.core import Dense, which caused module errors during runtime. To fix this, I updated the imports to use tensorflow.keras, which is the current supported version integrated with TensorFlow. This helped reinforce the importance of staying up to date with third-party libraries and having code evolve to meet best practices.