SNHU Capstone Project (Pirate Agent AI)

Noah Khomer

Southern New Hampshire University

CS 499 – Capstone Project

9/21/2025

SNHU Capstone Project (Pirate Agent AI)

The artifact that I selected for my enhancement project was the pirate agent AI project this was one of the things I developed in my artificial intelligence course the main purpose of this project is that it uses deep queue learning to train an agent to basically navigate through a maze and retrieve treasure

The first version of this project was implemented in Python as a single Jupiter notebook with reinforcement learning logic however one of the things this project really lacked at that time is that this professional structure means maintainability and lastly visualization.

Therefore, for this milestone, I made sure that I can enhance my artificial skills significantly. I turned it into a polished software project that can now demonstrate how advanced software engineering practices are developed with modular design, comprehensive testing, and interactive visualization.

I selected this hardback for my eBook Folio because it demonstrates my ability to design and implement different complex software solutions. These solutions are focused on combining algorithmic principles such as software engineering and user-facing design. The enhancements I made showcase A lot of different good software engineering practices.

The first thing I did was refactor the notebook into modular Python files. That means that now I have a Python file for train, models, maze and even configuration. The second thing I focused on was establishing the model view controller design pattern and this way my project was maintainable, scalable, and even professional in terms of file readability.

The second thing I focused on was adding full dark string inline comments. This looks good because it establishes a lot of different documentation standards by establishing A comprehensive README file so that it follows the PEP 8 coding standards.

I focused on testing and quality assurance by implementing a full testing suite with Test Treasure hunt Python file and this file. I tried to focus on unit and integration testing. Where I ended up achieving a coverage of 90%.

I focused on visualization and usability by establishing a GUI with Tkinter and adding real time training visualization using matplotlib which is a Python library. I can let the user watch the agent learn and analyze different performance metrics.

I also focused on a lot of different algorithmic enhancements. For example, I added the epsilon decay target networking experience reply so that it can really improve the training stability and performance.

I focused on creating a configuration system where I added the model saving and loading so that the results can be reproduced and even reused and this way I achieved reproducibility and persistence with coding standards.

With these improvements, I was able to turn the project from a basic academic assignment into an artifact that was demonstrating professional quality with software, design, engineering, and even the presentation.

Some of the things I discussed in module one was that I must design and evaluate computing solutions that can give a solution to problems using Algorithmic principles. So that the software solution can deliver value.

I was able to meet these outcomes by implementing a deep tune learning improvement such as epsilon decay, reward shaping target networks so that I can demonstrate strong algorithmic problem solving. Second, I focused on modularizing the code by building a GUI, writing the unit testing and configuration management so that I can apply professional software engineering practices. Taken directly from real-world settings. My enhancements fully aligned with the outcomes which prepare me to present myself as a capable software engineer.

Enhancing this artifact for my E portfolio taught me how I can use a sole use academic project and make it into a professional, reusable, maintainable software application. Some of the things I really focused on was the importance of that structure, making sure testing exists even in AI projects. I focused on the visualization for debugging purposes, therefore I do not have to rely on console alone. I made sure that there is balanced trade off with exploration, exploitation, and training when building the GUI. Overall, this process strengthens the ability to think, not just like a programmer, but even like a software engineer, these enhancements allow me to present the project in My Portfolio. That can demonstrate my technical ability to solve problems and even my professional growth.

References

García, Raúl Ferrer. “MVC: Model–view–controller.” *iOS Architecture Patterns*, 2023, pp. 45–106, [https://doi.org/10.1007/978-1-4842-9069-9\_2.](https://doi.org/10.1007/978-1-4842-9069-9_2.%20)

“Python Style Part 1 - PEP8.” *Computer Science*, cs.stanford.edu/people/nick/py/python-style1.html. Accessed 21 Sept. 2025.

“Reinforcement learning and deep reinforcement learning.” *Deep Learning in Science*, 30 Apr. 2021, pp. 282–307, [https://doi.org/10.1017/9781108955652.016.](https://doi.org/10.1017/9781108955652.016.%20)

Smythe, Richard J. “Realtime data plotting and visualization.” *Advanced Arduino Techniques in Science*, 2021, pp. 161–171, [https://doi.org/10.1007/978-1-4842-6784-4\_5.](https://doi.org/10.1007/978-1-4842-6784-4_5.%20)

“Unit testing.” *Software Testing and Quality Assurance*, 7 Feb. 2008, pp. 51–87, [https://doi.org/10.1002/9780470382844.ch3.](https://doi.org/10.1002/9780470382844.ch3.%20)