Enhancement 1 Narrative

This artifact is known as the Treasure Hunt Game from the CS 370 Current/Emerging Trends class here at SNHU. It is a machine learning project that had us implement a Deep-Q Learning process to train an agent known as pirate to solve a maze as efficiently as possible.  
 I selected this artifact for my ePortfolio because AI and machine learning is becoming increasingly prevalent as time goes on. That’s why I thought it would be a valuable addition to a ePortfolio. The reason I chose this specific artifact for all three of my enhancements was because there is a lot of room for change, improvement, and learning. For this specific enhancement, I successfully converted most of the code from python to C++. The only parts that weren’t translated were the python only Keres library for machine learning which I have temporarily had to replace with a dummy model and the maze drawing function that was used once to show the pirate agent’s route and, in my opinion, is not a necessary part of the new version in C++ so I therefor didn’t include. Unfortunately, as of this moment, until enhancement 2 this artifact will not be able to run since that is the enhancement where I plan to implement the model and multi-Agent training formula enhancement. The dummy model, which is a place holder, doesn’t possess the necessary functionality for this near one-to-one translation from python to C++ to work. However, despite the artifact not being able to run until the next enhancement, it has been improved since C++ is better at low level management and therefor should have increased efficiency and speed when training the future model. Additionally, the current code is far more up-to-date and therefore somewhat easier for others to implement and use.  
 Overall, I believe I met the course outcomes I planned for this enhancement. However, this entire process has surprised me at the difficulty of it all. I’ve had to invest more than a little time in finding equivalent variable types and creating classes and structures capable of handling the functionality of the original artifact. And not only that, I found myself constantly stopping and looking between what I was writing in C++ and what was already done in Python. This entire ordeal was definitely challenging, and I feel like I could do something similar again with more confidence and efficiency since this was the first time I’ve done something like this.