**Milestone Three**

**Enhancement One: Algorithms and Data Structures**

1. **Briefly describe the artifact. What is it? When was it created?**

The artifact I chose to use is one of the first projects I completed when I started my journey at SNHU back in January or February of 2023. It is a text-based video game that I developed using Python in IT 140: Introduction to Scripting. The goal of the project was to develop a text-based video game that allowed a user to enter the text “Go North, Go South, Go East, or Go West” to navigate through a maze of rooms that was provided to us. We were able to decide the theme of the game, what each room would be, and what items would be available to collect in each room. Each room only had certain directions available depending on the direction of the openings displayed in the provided map. The player had to visit each room and collect an item to have everything necessary to kill the boss in the final room. If the player failed to collect all the items they would lose the boss fight, otherwise they would succeed.

1. **Justify the inclusion of the artifact in your ePortfolio. Why did you select this item? What specific components of the artifact showcase your skills and abilities in software development? How was the artifact improved?**

I chose this artifact because it began as a simple project with a lot of potential, and I knew that with more development knowledge and skills, I could expand it significantly. Even in its original form, the project demonstrated my ability to code in Python, write modular functions for specific tasks, and create well-organized, well-commented code that was easy to follow. I enjoyed building it and wanted to enhance the user experience by making it more engaging and interactive. To satisfy the Algorithms and Data Structures category, I expanded the project’s complexity by adding a graphical user interface using WinForms and integrating a database to allow players to save and load their game states. I also implemented a secure hashing algorithm to protect player passwords, improved the MovePlayer method for more efficient handling of UI button clicks, and optimized the WinForms interface by reusing controls to improve performance. Creating the WinForms interface showcases my ability to design a functional UI that dynamically hides or displays controls as needed. Integrating the database demonstrates my skills in building an SQLite database connected to the UI and developing methods to manipulate data directly through the interface. Additionally, my object-oriented programming skills are reflected in how I restructured the entire project into multiple classes, applied encapsulation, and transitioned from standalone functions to methods within internal classes, each serving a specific role in the overall functionality of the game.

1. **Did you meet the course outcomes you planned to meet with this enhancement in Module One? Do you have any updates to your outcome-coverage plans?**

Yes, I met outcomes 1 and 3 as I initially planned in Module One with my enhancements for the Algorithms and Data Structures category. To address outcome 1, I organized the code to be maintainable, easily adaptable, and well-commented so that it is easy for other developers to understand and collaborate on. I also followed meaningful naming conventions for methods and variables to improve readability. For outcome 3, I used dictionaries for efficient room navigation and to map items to their corresponding rooms. Additionally, I implemented a password hashing algorithm to protect player passwords and created an SQLite database with methods to store and manipulate game data. At this point, I have achieved the outcomes I originally targeted and do not have any updates to my outcome coverage plan.

1. **Reflect on the process of enhancing and modifying the artifact. What did you learn as you were creating it and improving it? What challenges did you face?**

While working on the enhancements for this category, I faced several challenges and gained valuable knowledge along the way. As my project grew and the codebase expanded, it became easier to overlook mistakes and encounter bugs. Even though the project may appear simplistic, ensuring that the form behaves correctly, the game logic functions properly, and the database interactions work as expected has been challenging. One issue I faced was that after hitting a wall in the game, clicking the “Go Back” button would always return the player to the Main Foyer instead of the previous room. After troubleshooting and reviewing my code, I discovered that I had mistakenly named the method parameter and the private class variable both as currentRoom, causing unintended behavior. Renaming the parameter to newRoom resolved the conflict. Additionally, I realized I had unnecessarily created two separate objects, a game object and a player object, and was referencing the wrong one. I removed the extra player object and updated all references to use GameInstance.Player.\_\_\_, properly using the player instance within the game object. Through this process, I learned how to create objects of one class within another and correctly call their methods, which is something I hadn’t done before in my previous projects.