Category 1 Enhancement – Software Design/Engineering

Miguel Little

CS-499 Computer Science Capstone

Professor Krupa

08/07/2024

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

I selected a Vector Sorting project from my CS-260 Data Structures and Algorithms class. It was a basic project built on C++ that imported bids from a local CSV file. Once the bids are imported, the user can sort the data with a few quicksort functions utilizing basic algorithms. The bids would get sorted using loops that sort by highest and lowest bids per loop iteration until the whole list is fully sorted.

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 included this artifact in my ePortfolio because it was the most diverse project I worked on in my time here at SNHU. It allowed me to plan three enhancements that met each category within the same project. This was an opportunity to build an application reflecting a real-world project for this career. By creating sort-by-parameter functions for this project, rebuilding it in Python, adding a functional GUI, and making it modular, I showcased my skills and abilities in software development.

I improved this project visually by implementing a basic GUI in Python that parses the data displayed in that chart. I also enhanced the project's functionality and usability by creating classes to make the application modular, practical sorting functions, and an “add a bid” function. I removed redundant code, such as the “close now” button, since the GUI now had a close button built into the window.

I implemented merge sort to the sort by closing dates functions since it is on par with heapsort. When adding new bids the application now checks the code and ensures that the added data is correct. The bid amount must be a REAL number, the bid ID must be unique, and the date fields must match MM/DD/YYYY.

I also added Unit tests to test the code functionality. The unit test that works the best checks the article ID and ensures it is Unique. The GUI now doesn’t show the “Enter a new bid” box unless “Insert a bid” is selected and submitted.

This application can be built into many different businesses that require data logging and sorting; each sort function is a specific component that can be used in the field. GUI implementation is also a component many businesses can implement.

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

I will list the outcomes I met in this assignment, how I met them, and what my plans are:

The first outcome met: ***“Design, develop, and deliver professional-quality oral, written, and visual communications that are coherent, technically sound, and appropriately adapted to specific audiences and contexts.”***

I met this outcome by converting the project into a Python application and implementing a basic GUI. I then commented on each function and added headers to each file. To take this application to the next level, I modularized it.

The second outcome met: *"Demonstrate an ability to use well-founded and innovative techniques, skills, and tools in computing practices for the purpose of implementing computer solutions that deliver value and accomplish industry-specific goals (software engineering/design/database)."*

I implemented the GUI and used event handlers for the radio select buttons to manage the sort-by functions. Each sort-by function utilized improved sort algorithms. Adding a bid function and the sort functions made this project applicable to many business models, making it industry-specific.

The GUI only shows the "insert a bid" entries once it is selected and submitted; before, it would show constantly, which caused confusion. All of the inputted data is tested and validated with error handling in place. Unit tests ensured that each article ID was Unique.

The third outcome met: "Employ strategies for building collaborative environments that enable diverse audiences to support organizational decision-*making in the field of computer science."*

By uploading this project to GitHub, adding a readme file, and commenting out each section of the code, I made it collaboration-friendly. Other coders can easily jump in and build on it.

I will meet more course outcomes when I add time complexity and improve the database implementation in this code.

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?

Reimplementing each of the application functions was difficult in Python. The second biggest challenge was giving the code an object-oriented design.

The third problem was very tricky: the memory would get deallocated, which was fixable by implementing the built-in copy function that created new memory sources.