Matthew Cohen

To begin with, I must admit that I thoroughly underestimated the sheer amount of kerfuffling that these projects would require and by my own admission believe that I may have gone a bit far in some areas.

In the interest of brevity I shall simply list off all the learning outcomes here and refer to them only by their number in my narrative:

1. Employ strategies for building collaborative environments that enable diverse audiences to support organizational decision making in the field of computer science
   * Does the student demonstrate the ability to use interaction to create code-review experiences?
   * Does the student demonstrate the ability to understand code reviews individually and within a team environment?
   * Does the student demonstrate the ability to provide contextual, in-code comments that result in easily readable and understandable code?
   * Does the student demonstrate the ability to support decision making for software design stakeholders?
   * Does the student demonstrate the ability to discuss experiences and best practices working in collaborative environments?
2. Design, develop, and deliver professional-quality oral, written, and visual communications that are coherent, technically sound, and appropriately adapted to specific audiences and contexts
   * Does the student demonstrate the ability to communicate a code review effectively through collaborative environments?
   * Does the student demonstrate the ability to discuss experiences and best practices in communication?
   * Does the student demonstrate the ability to communicate appropriately to specific audiences and contexts?
   * Does the student demonstrate the ability to clearly convey their ideas and explain their thought process through written, visual, or oral communication?
3. Design and evaluate computing solutions that solve a given problem using algorithmic principles and computer science practices and standards appropriate to its solution, while managing the trade-offs involved in design choices
   * Does the student demonstrate the ability to use pseudocode to segment functionality in software and make design trades?
   * Does the student demonstrate the ability to program solutions to solve logic problems and implement them in software?
   * Does the student demonstrate the ability to clearly articulate approaches to solving complex logic problems inherent software?
   * Does the student demonstrate the ability to discuss experiences and best practices in designing and evaluating computing solutions?
4. 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
   * Does the student demonstrate the ability to employ iterative testing techniques in the code?
   * Does the student demonstrate the ability to use the software development life cycle to create realistic production schedules for software projects?
   * Does the student demonstrate the ability to create industry-standard software designs?
   * Does the student discuss experiences and best practices in using well-founded and innovative techniques, skills, and tools in computing practices?
   * Does the student create more robust and efficient code to deliver value and accomplish industry-specific goals?
5. Develop a security mindset that anticipates adversarial exploits in software architecture and designs to expose potential vulnerabilities, mitigate design flaws, and ensure privacy and enhanced security of data and resources
   * Does the student demonstrate the ability to address potential design flaws in software architecture during the requirements phase?
   * Does the student demonstrate the ability to find and eradicate security vulnerabilities or bugs?
   * Does the student demonstrate the ability to define an approach that ensures all data are explicitly validated?
   * Does the student demonstrate the ability to consider future changes to objects and classes?
   * Does the student demonstrate the ability to discuss experiences and best practices in developing a security mindset?

Enhancement 1:

This enhancement was by far the more time consuming of the two; initially I had planned to simply refine the existing code. The original project was a fairly rudimentary renderer of a very basic and rather hastily put together 3d model of a house living room and my initial proposal was to expand and add detail to said living room since the initial rendering really wasn't very good looking or detailed. However, this is not, you might find upon review of enhancement 2, what ended up happening. It was brought to my attention by a friend that I often seek out help from that realistically the method of rendering used in the initial project simply wouldn’t be used for anything of the sort in any actual use case due to depreciated code and methodology, the overall construction and setup of the project was not the greatest (to put it mildly), and actually adding to it via making 3d shapes myself via code simply wasn't feasible if I wanted to end up with a result that was even close to looking good, much less professional. Additionally, I truly am not a fan of C++ so the excuse to “port” the project into C# instead was greatly appreciated.

With all of this in mind I set about learning about a different and allegedly more proper way of handling the task of doing a render like this and eventually got a firm enough grasp on moonworks (which is a free to use software for this) to create an entirely new program which renders imported 3d objects in a far better (I don’t necessarily have the space or the time to go in detail into all the ways in which it is) manner. The end result is that of a substantially better looking room and objects within as well as a substantially more modernized codebase.

Overall I am far more satisfied with this as a demonstration of my abilities and also like the end result visually more too; having to essentially learn moonworks and how to get it to function and connect was a major learning experience for me and it was certainly touch and go for a bit, however, in the end it really helped me to grow as a coder. I think that this project aptly fits learning outcomes 2, 4, and 5 due to the fact that this new project looks much closer to something that could actually be implemented into something without looking notably awful, demonstrated my ability to learn and use contemporary tools like moonworks in order to reach a satisfactory result, and had a code structure and overall design that is far closer to something that one might actually see in practice as opposed to something just done as a part of school project.

Enhancement 2 and 3:

For these enhancements the original artifact is a program created as part of my data structures course which would search through and return an entry from a csv based on user input. Users would input the course number and said course would be returned. This was a fairly natural inclusion since its original form was quite limited by what you could search for courses by and also simply read courses off of a csv instead of a database.

For this enhancement too I determined that I simply didn’t like C++ very much and that ultimately if I were to be hooking it into a database anyways then it would be best to simply start from scratch with an entirely new codebase and language. Once I was done the enhancement greatly increased its usability and took steps to potentially allowing it to be feasibly used since users would likely not want to download the csv every time and classes would have to be updated by administration, both of which are demands that can only be met with a database.

I attempted to meet the course requirements 2 and 3 and given the fact that enhancement 2 made the program actually able to perform tasks that users would want and enhancement 3 made the program actually able to be used in a software ecosystem that could exist I think that said enhancements did meet said course goals.