Jeremy Bridges

CS 499

6/19/2020

The capstone course has allowed me to reflect on all that I have learned at Southern New Hampshire University. I have reviewed all my completed works in the computer science program and chose 3 to enhance and present as a representation of my skills. During the code review process there were minimal issues found in the original projects which shows that my programming skills were strong throughout the courses I took. In one of my first SNHU courses, “computer science role in industry”, I would look to go above and beyond on simple assignments so I would learn more than was required. I had an exercise to create a menu of services for an automobile repair shop and used a dictionary to store the different services and related costs rather than just a simple list. Dictionary objects were not required but I had recently learned about them and they worked very well in the solution.

One skill I have exhibited strongly throughout my education, has been the ability to collaborate in a team environment. Every course and every week I would check the discussion boards and look to give and receive help from others. I was always willing to offer advice if I understood the concepts or request help from others if I did not. The team concept is relatable to me since I actively enjoy playing team sports. When we are in the same class or same work group, I look to help the team succeed, not just myself.

Communication with stakeholders is an important skill I have begun to develop while at school. It is important to use tools and methods which support strong communication between the team, management, and the customer. Code reviews can be a valuable tool to keep others up to date with what is going on with a project.

A strong understanding of the different data structures and algorithms that can be used across many different languages is necessary to be a successful in computer science careers. Choosing the correct data structure is vital to creating efficient and effective programs. Successful software engineering encompasses understanding what the software requires and then analyzing, designing, and testing until it meets the customers requirements. Security should always be kept in mind when working on any project. Defensive programming requires testing input and output for validity and completeness. It also requires making sure memory is allocated and deallocated properly as well as file usage bounds. It is important to not leave this till the testing phase but rather and ongoing process to keep programing secure.

The first artifact I have included is a visual design program in C++ language using the OpenGL API library. This artifact is the best example of my software design and engineering capabilities. This is by far the largest and most complex program I worked on while at school. The project had many working parts including matrix mathematics, shader programs and source code, buffer and array objects, camera objects and input callbacks. This project required a lot of learning quickly to complete the requirements.

The second artifact I have included is an authentication system which was initially written in Java but was converted into Python for this portfolio. Although the program is much smaller than the first artifact, it required solid understanding of data structures and algorithms. The program involves using a user defined class object, hashing, nested loops, and multiple conditional statements. It required strong organization and logic comprehension to keep all the different parts working as intended. Converting the program from Java to Python required new learning but also to understand the program from a non-specific language context.

The third artifact I have included is a HTML/JavaScript interface that uses Python and Bottle to create a RESTful server that can access a MongoDB database. The HTML/JavaScript interface required more research since we did not learn much about it during the course. Working with databases has increased my interest and consideration of a possible career within computer science. Database manipulation primarily involves the create, read, update, and delete (CRUD) operations but there are many ways to implement those operations. This artifact allowed me to learn a new way to interact with a database. Enjoy the following examples of my best work to date.