Milestone 2

Chad Shafer

SNHU

The artifact that I chose to represent is a small game that everyone knows which is Hangman. The code is written in C++ and was created a year or so ago. The program is composed of 7 functions and a main function. The first function is called printMessage. This function sets up the borders to the game and it called when the game wants to print out certain words. The next function is one that I recently added in. This function is called DrawHangman. Its purpose is to print a piece of the hangman every time the user enters an incorrect guess. The third function is called PrintLetters. This function is set to print a list of letters ranging from one character to another, leaving off characters that the player has already guessed. The fourth function is called PrintAvailableLetters. This function prints the letters for the player to guess. The fifth function is PrintWordAndCheckWin. This function prints the word the player must guess. It shows the letter if the player has already guessed it and shows and underscore if the letter remains unguessed. The sixth functions is named LoadRandomWord. Just like the name suggests, this function loads a file containing a list of words and picks a random word to use for the player to guess. The seventh and final function before our main is called TriesLeft. This function determines how many incorrect guesses the player has made. Finally, is our main. This is where our program is run and calls to the other functions. When working correctly, the user has a fun little hangman game.

I used this artifact because I feel that this program, along with the enhancements, is a good representation of the range of skills needed to perform as a coder. I believe that this code demonstrates a combination of knowledge, aptitude, technical capabilities and problem-solving when it comes to developing a working code based of user’s expectations. While I have utilized different coding languages such as Java, Python, and SQL, I prefer to work with C++ as it seems most appropriate to the field of choice. When it comes to the software design of this program, I needed to create a design that met the users’ requirements. When you look at the requirements, the game needed to be the traditional functional, hangman game that everyone knows and loves. This program demonstrates the problem-solving aspect of software design and demonstrates how I can take the problem to the solution. The different functions define the components to define the behavior and boundaries that I wanted in the game. I made some enhancements and that proves that my code is flexible and able to be modified on the changing needs. In the case of design and engineering, I lacked the drawing of the stickman when the user guessed the wrong letter. I developed a function that drew the stick man in on a wrong guess. I feel that this enhancement meets the users’ expectations more than the previous version. I have also cleaned up the code to make if fall under some best practice coding. This creates the clean code approach which is easy to understand and easy to read. Before my functions were practically connected, but now there is adequate space. The last enhancement that I put in place was to accurately comment my code. This details what the code and functions are doing. I feel with these enhancements in place, this program I have met the course objectives.

I can honestly say that when I was first putting this code together it was a struggle with a lot of research and help from my brother who is a software engineer in Michigan. In the beginning it felt like running on a treadmill. You are running fast but going nowhere. That’s what this journey through this major has felt like. However, when I look back at this code for this course, I was able to see where enhancements were needed and what I could do to make the program better. To me that shows that there is growth and when I felt like I was going nowhere, I was moving more than I thought. There is still a lot I do not understand when it comes to programming but there is always information to learn from. The commenting and clean coding were the easy part to the enhancement, but it does enhance the code greatly. The harder part will come with the algorithms and data structure and working on the seeding randomizer.