CS 4900 – Quad Solver Stories Report 5 (Pre-Release)

Steven Johnson

Brandon Rodriguez

Joshua Sziede

11/05/2018

1. **Description**
   1. We are writing a program that can take a given user input of numbers and factor their quadratic, as requested by our client, Dr. Kapenga of Western Michigan University, using the common quadratic formula.
2. **Requirements**
   1. Language
      1. The program will be written in the C language using the Carnegie Mellon University C coding standards and the GNU compiler collection.
   2. Platform
      1. The program is intended to run on machines using a Linux distribution. Testing will occur on machines running the latest versions of Ubuntu and Mint.
3. **Stories**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Story** | **Time Estimated** | **Risk (out of 5)** | **% Complete** | **Total Time** |
| The user is expected to input a single string that contains three numbers *a*, *b*, and *c*, as a command line argument. Equations are expected to be in the format *ax^2 + bx + c = 0*. This string will then be parsed and split into three numbers in a normalized, 32-bit, single precision data type. If the input string is not provided as a command line argument and/or the input cannot be parsed into three numbers, the program will assume no input and ask the user to input a new string of numbers. | 2 weeks | 3 | 99% | 6 hours |
| Output *x* is calculated by the quadratic equation using *a*, *b*, and *c* as a normalized, 32-bit, single precision data type. If *x* is a real number, we show the number to the user using the terminal. If *x* is an imaginary number, we discard *x* but let the user know through the terminal that an imaginary number was calculated. If there exists a possible rounding error for *x*, the program will also alert the user through the terminal. | 1 week | 2 | 100% | 0 hours, 30 minutes |
| Argument parsing will occur in its own source file with its own header file. | 1 day | 1 | 100% | 2 hours |
| Miscellaneous helper functions are also put into a single source and header file. | 1 week | 1 | 100% | 1 hour, 30 minutes |
| The solving of the quadratic equation will occur in the main source file. | 3 weeks | 3 | 100% | 5 hours |
| A makefile exists to provide shortcuts for compiling, debugging, and testing the program. | 1 day | 1 | 100% | 1 hour |
| Input reading, input parsing, and output printing are each performed separately from one another within their own functions. | 2 weeks | 3 | 100% | 4 hours |
| Calculations will be performed within the main source file. | 1 week | 2 | 100% | 3 hours |
| A spike will be performed on IEEE single-precision floating-point numbers, make files, stdin, and logging. | 1 week | 2 | 100% | 4 hours |
| Unit testing will be performed using CUnit. | 2 weeks | 1 | 75% | 5 hours |