CS1220 – C++ Programming

HW#8: Term Project. Due: 4/27/22

Points: 150 (30 + 120)

Name: Joshua McKinniss & Christopher LaFave

I. Requirements: Restate the problem specification, and any detailed requirements

Create a program that reads in a circuit file "____.txt" and a vector file "______v.txt" and outputs a simulation of the circuit.

II. Design: How did you attack the problem? What choices did you make in your design, and why? Show class diagrams for more complex designs.

We broke the problem up into parts: parsing the circuit file, parsing the vector file, simulating the circuit, and then outputting the history. To do that, we needed to make and include 3 classes: Wire, Gate, and Event.

III. Implementation: Outline any interesting implementation details.

One of the cool implementations we did was how we filled in the history vector for each wire. When a wire is used in an event, we update the history of what is behind the change that was just made. For example, if A starts at low and goes high at 4, when that event pushing it high goes through is when we fill in low for 0-3, and then set 4 as high. Then when the Events queue is empty, we fill in the last bit of history up until the simulation stops.

IV. Testing: Explain how you tested your program. Explain why your test set was sufficient to believe that the software is working properly, i.e., what were the range of possibilities of errors that you were testing for.

Whenever we created code that compiled for each step, we would take all the example circuit files and run them to make sure they worked. When they didn't work, we would look through the code to see why it broke and then fix the bugs until all the example files worked. Once we got working code for that part, we'd move on to the next one.

V. Summary/Conclusion: Present your results. Did it work properly? Are there any limitations? If it is an analysis-type project, this section may be significantly longer than for a simple implementation-type project.

Yes! It works properly. It outputs what we believe is the right output for all the circuits we tested and has the correct error handling measures.

VI. Lessons Learned: List any lessons learned. What might you have done differently if you were going to attack this again?

If we were to do this project again, we probably would start earlier and use GitHub. Because Chris had his computer in IT for a long time into the project, we had all the code on Josh's computer and decided to not use GitHub. That decision made it difficult for us to work independently. We also would start the project earlier, as completing this project came at the cost of working for multiple hours straight on this project to compete it before the deadline. Another good lesson learned is how to better utilize Visual Studio's debugging features. When we would have a circuit that didn't work then we would set breakpoints, look at the call stack, and set variables to check what they were at the point of error. Using this technique, we were able to fix all the problems that we came across.