CS1220 – C++ Programming

Homework Assignment #5
Due: See course web site for due date
Points: 80

Name: **Sam DeCook**

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

Create a tic-tac-toe game using wxWidgets. Include a menu, a status bar, and reset and exit button.

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.

Much of the GUI set-up was explained for us. I didn't have enough time to explore wxFormBuilder and make my game look good, so I just put the reset and exit buttons right below the board.

For the game logic, I ran through some pseudo-code of what happened on each turn. I turned most of those into functions and created the needed data members to run the game. I had already programmed tic-tac-toe before, so it wasn't very hard.

III. Security Analysis: State the potential security vulnerabilities of your design. How could these vulnerabilities be exploited by an adversary? What would be the impact if the vulnerability was exploited?

I don't know if there are any. The only input it accepts are mouse-clicks. Maybe you could hack the OS to put in your own mouse-clicks, but at that point you're already in the OS, why attack my silly little tic-tac-toe game?

IV. Implementation: Outline any interesting implementation details in your solution.

For the win detection, I ended up going very basic. I had a big if statement where I checked each possible combination, and OR'ed them together. While it is basic, it is the shortest way to do it; however, and I just thought of this, checking for a win with the last token placed would probably be the most efficient method.

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

I tested it by checking all of the possible win cases with X. Because I built the CheckWin function with the player token as a parameter, it will (and did) work with O. finally, I made sure all the buttons worked properly.

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

It worked properly. The only limitations are visual; I only had time to finish the logic part and not work figuring out how to change the layout.

VII. Lessons Learned: List any lessons learned. For example, what might you have done differently if you were going to solve this problem again?

Don’t try to download windows or even linux-based software onto my Mac, it just doesn’t work. The main lesson I learned this time was to make sure that every button has the OnClick… function in its event tab, I spent a long time trying to figure that one out. I would also spend some time trying to make the CheckWin function as efficient as possible, it seems like a cool problem to try to solve.