

CPSC 3720 – Assignment #2 Report

1. What was the easiest aspect of extending another developer's software?

The easiest aspect of extending another developer's code is that they have a code already there to reference the code if ever implementing another derived class; which makes it more easier to start with when doing code extensions. Also small mistakes on the GoFishTest.cpp where most of the cards are all clubs, so I made it what suit they are assigned to.

2. What was the most challenging aspect of extending another developer's software?

Sometimes, when going through coding for implementation and testing, there are some functions in the base class that are needed to be virtual, because the process of the function maybe different from the new derived class. Also, maybe one of the existing functions for the existing derived class may be redundant for the newly implemented function (for example. choosePlayer from GameUI.h). I had a problem doing tests for the new code extension, and it took me a while till I figure out it's a Makefile problem: On that file from lines 7 – 9, SRC is used to designate to test compile all "src/*.cpp" without main.cpp but in result not compiling with new cpp files. So I had to figure out the line of code in Makefile for a way to compile everything with newer .cpp files excluding the main.cpp file.

3. How would you rate the quality of the code that was provided for extension?

Provide a justification for your assessment.

Between 0 to 10, I would rate it an 8/10 since I am able to compile extension, but I wasn't able to figure out how to make some classes in GameUI like choosePlayer for GoFishUI or (newly created) listPatternUI for Rummy not virtual, because they are all connected with GameUI. Rummy.h and GoFish.h aren't connected with their respective view headers, making some of the virtual functions meant for the specific derived classes be redundant for the other one. I wasn't able to do all test for the new code extension, but since it's almost the same as GoFish, it made it a bit more easier to do tests in a similar manner.

4. What could the previous developer have done to make new feature development easier?

I wouldn't have made the list pointers because I have encountered many memory issues (majorly Stack Overflow) throughout the implementation. Keep any array-based containers out of pointers and keep them as it is. Also, I would make sure all the classes defined will be in the header files, and make sure to use them. And make sure to check your Makefile if it can compile files that includes new ones.

5. What could you have done with your software that would have made the job of the maintenance/new feature developer of your software easier?

I would make the derived controller classes more connected with the derived viewer classes (UI), rather than just only the base controller class have more lenient connection with the base viewer class since some functions are exclusive to variety of derived classes.