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

- The easiest aspect of extending the code that was given was that I did not have to create the models for the controller. This made it easier because I did not have to write the code for the models, saving me time on that aspect of the assignment. Another aspect that made the code easier was the implementation of GoFish allowing me to see how it was using Game and GameUI and trying to emulate that for Rummy and RummyUI.

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

- The most challenging aspect for me for extending the code I was given was the lack of multipurpose functions and/or functions for everything that is needed for a cardGame that is *somewhat advanced*. The functions in Game and GameUI that were given, barely allow you to create a simple cardGame, let alone a cardGame like Rummy. I needed to add many private functions in Rummy and added plenty of new functions to RummyUI that were needed to be static casted in Rummy. This was needed because I did not want a very ugly and not fully functional game of Rummy. You would think just reading the code and understanding it would be a challenging aspect, but I have a lot of experience in helping others and quickly reading their code for finding errors or adding optimizations.

How would you rate the quality of the code that was provided for extension? Provide a justification for your assessment.

- I would rate the quality of the given code a **2**. This is because the code requires so much extra work to implement a game like Rummy. This is because the main Start() loop for game has basically two functions that can be used for the game: **beforeCardPlayed** and **afterCardPlayed.** For my assignment 2 I wanted to go the route of not changing the original code at all (which I accomplished). The only thing I added was the destructor for deck. Since this was the case I had to implement the entire game of Rummy in the two functions **beforeCardPlayed** and **afterCardPlayed**. I also ignored requestCard because in Rummy you do not ask players for cards. The code that was given was basically created for the most basic cardGame or specifically for their implementation of GoFish. Also, the tests and code had memory leaks I had to fix.

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

- The previous developer could have added more multipurpose functions to Game and GameUI to make new feature development easier for the implementation of a new game. The lack of functions to used in the Start() loop that runs the game makes it so the code has to be implemented in entirely **beforeCardPlayed** and **afterCardPlayed.** Since the GameUI lacks functionality for an advanced game like Rummy it requires the controller (Rummy) to static cast its UI to RummyUI since I am unable to implement Rummy using only the functions that are given in GameUI.

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

- Many of my functions seem specialized for a specific task in Rummy, maybe it is possible that I could make a more multipurpose function that could reduce the amount of code. Another thing that I could do is increase the amount of comments used for explanation in my cpp files. Lastly since I did not complete my tests, finishing those for complete proof of functionality would help with maintenance/new features.