Team 19:
Beatrice Villanueva, Jeffrey Shao, James Gruber, Shandy Sulen, Charles Griffin, and William Benda

Final Project Report

Lessons Learned

What problems did you encounter (timing, communication, implementation, merging)?

Villanueva - Personally, the biggest problem I encountered was the conversion between the Solitaire logic developed by Sulen and Gruber into the code utilized by the Qt GUI Software. We had not anticipated the learning curve with utilizing the GUI, and a lot of time was spent first trying to understand and also trying to implement Sulen and Gruber's code correctly.

Sulen - Learning the implementation of new software was the most challenging obstacle to overcome. Jim Gruber and I personally worked on the back-end side of the project and wrote the .cpp source code for creating the logic behind Solitaire. We both learned Git and Github for source version control and merging code. This process was frustrating as we were new to the software, but slowly we learned the git commands as well as the solutions to the errors that we encountered.

Gruber -The largest difficulty on my part of this project was dealing with such a large task at hand. Up until now, my assignments were normally not much larger than ~500 lines of code. In this project, Sulen and I ended up writing about 2000 lines. The amount of data and the logic implemented was more complex than most I have had to deal with in other assignments. I learned that it helps greatly to plan ahead and figure out how I will implement certain logic before I even get there.

Shao - Learning to use a UI was the steepest learning curve I think all of us working on front-end faced. After 2 years of working strictly in the console, it was difficult learning all of the UI-specific syntax and the logic that is used to make UI's work. Qt (the GUI software we used), while powerful, was very non-intuitive to someone who had never used it before. There was also much difficulty managing all 52 cards, and making sure that they would do the right thing under every possible circumstance that can possibly occur in a game of Solitaire.

Benda - I was the team member that first found and began learning how to use the IDE. Learning QT was definitely much more difficult that I originally anticipated. It required many more hours of work to get a functioning solitaire game. The team dynamic worked well, and through our combined efforts, we created what we did.

Griffin - The biggest problems occurred with synchronizing the backend code with the Qt specific syntax. When we were trying to connect button movement to the nodes in each of the vectors, it was tough to make sure they were working together. That way, the backend code that

was checking the logic of the code correctly identified which cards were being moved and where they were being moved to. I, especially, had problems implementing the rules and moving whole stacks of cards at a time. I didn't realize how much goes into making sure the user's don't mess up the movement of the buttons and stacks. We spent a lot of time meticulously testing the final program to test for bugs, many of which were especially finicky. Also, we used a click-to-click method for moving buttons, which was finicky if the user wasn't careful with making sure the clicks were in sync.

Future Work

How do you plan on using this experience for future work? Do you plan on continuing working on this project or a related project?

Villanueva - I will definitely use this experience for future projects in any language and in any GUI software. Now knowing the learning curve involved with GUI's, I will plan ahead and dedicate time to testing the GUI for what we need as well as plan for how the hard code will merge with the front end code. I also plan on continuing to work on this project and expanding its functionalities. I would like to see what else Qt can do and what it could potentially help me with in future projects. That being said, I am really proud and thankful for this group of people. This project would not exist without all of their help and contributions; Our communication, dedication, and persistence was very admirable, and I would gladly work with them again

Sulen - This experience taught me the value of teamwork as well as the process of designing the functionality of code before even writing a single line. I will utilize the insight I gained from working with Git and source version control as well as communication skills in the future projects I take upon myself. I am unsure if I would ever develop more functionality for the Solitaire project, but I absolutely would enjoy working on more games in the future, perhaps even for other platforms like iOS or Android.

Gruber - This experience will help me work with others and realize that big projects are possible. All we have to do is make a plan and dive into the coding. I have also learned that certain logic is difficult to see, and having multiple people working on the same problem helps lead to success. I probably will not work on this project, but my peers' ability to use QT and GUI's inspires me to learn them as well. When it comes to projects like these, you need to be able to contribute, and being versatile in multiple tools makes contributing significantly easier.

Shao - The experience taught me how meticulous and systematic you really have to be when programming. Something as simple as moving a card from point A to point B took hours on end to figure out. I also learned how complicated things can get when you throw in GUI's, and hope to polish my skills on those. I doubt I'll touch this project in the future, but I do hope to continue

working with GUI's so that I can provide visuals for all of my code, whether that means learning the ins and outs of Qt or moving onto a different language/UI software.

Benda - This experience gave insight into team projects. Managing resources, clarity, and focus are all important in such endeavours. We had great communications skills, and a common understanding of what we sought to accomplish. If we hadn't had such unity, we would have gotten infinitely less work accomplished. I cannot say that I would use Qt again for such large projects, but it would be useful for smaller scale ones.

Griffin - This experience was extremely enlightening. The amount of work that goes into creating a fully functioning game is enormous, and it was really important to keep things clear for other members of the group to work off of. This was especially important when converting the console game to the GUI. We were able to make use of a lot of code and logic because of the clarity of the backend code, which was great. For the future, I don't think I would use Qt because it wasn't especially friendly with data management and manipulation of QWidgets. This project definitely fueled my interest to continue working on larger scale projects, though. I'm extremely interested in working on similar projects on different platforms like iOS, and expanding my experience creating GUIs.