Name: Louis Hwang

Term: Fall 2023

Previous Team Projects

I have been studying computer science for a little bit over a year since I started studying at Oregon State University. Therefore, I do not have very much experience with working as part of a programming team. But there were some classes that involved a short group project, and one of them was in CS 162: Intro to Computer Science II. In this class, one of the group projects required group members to review everyone else's code for a certain assignment and then as an entire group determine who had the best implementation of the assignment. There were no defined roles for these kinds of group projects, so I always attempted to act as the initiator, where I would start the discussion with everyone and have all the group members give their opinion on whose code was written the best. I feel that I should play this role because I am often the most motivated member of the group for CS 162 and it does not feel right to just wait until some else initiates first when I am in a position to do exactly that.

In a purely online group format, the main difficulty that arises from this format comes from when one or more group members simply do not communicate in a timely manner when necessary. Although I personally did not have this issue when I worked on group projects, I have heard of other students who say their group members are unresponsive and end up having to do all the work themselves. I feel the cause of this problem is due to some group members being too busy in their daily lives and forgetting that they need to get involved, and the fact that the group project is often largely asynchronous is a big factor for causing that. I would generally just do all of the work for the group if that situation were to happen to me since I have only worked with groups in an academic setting so far, but I understand that in the real world, the work may be too much for me to do all alone, which means that establishing proper means of communication is even more important once I move past working with groups at school.

Working with Continuous Integration

I thought Continuous Integration was a very interesting way for group members online to work on one project. Initially, I was not sure if Continuous Integration would work well for this project since I was not sure how to set it up at first and since my group members would all be working on this project asynchronously. But I was able to overcome these initial reservations because the directions in the modules were very clear about how to set up the environment needed for the group project. In terms of how everyone would work on the project, I decided that since everyone would be working on the project at different times and since we did not expect to hold any meetings, I felt that each group member should work on a function of their choice until the task is completed and then have other group members review the code before integrating the code into the main branch. This seems to be the most logical approach given the online nature of my school. If this was an on-campus environment, then I would probably take the approach of meeting together and working together on each function one at a time so that everyone can learn some programming techniques from each other.

The code review process was helpful for all of us in the group since feedback allowed us to make changes to our code and write our code in a different way. For the entire project, I think my group

members did well with writing their functions and adding relevant tests to make sure we can obtain the best score possible. Therefore, besides asking other group members to fix some linting errors and other small errors that sometimes occurred, my group members and I would often just make stylistic recommendations in our Code Reviews. We generally just made pull requests whenever we felt that our part of the code was ready for integration, and although that may not be as effective as doing daily commits or more frequent commits as described by the Test Driven Development Process, I felt that the way we handled pull requests was the best way to do it given that we worked asynchronously and that each of us spent our time just doing one separate function per person rather than having everyone do the first function and then the second one and then the third one. Therefore, given our circumstances, I think everyone was able to work well together and complete the group project by the deadline.

As a developer, I feel that I had learned a lot from working with Continuous Integration in this project because I never previously thought about working with projects in this way. For example, I initially wasn't sure how different group members could merge different parts of the task together into one file, but since I was the one to handle the merging in this project, I was able to learn firsthand how the merging works. As a team member, I learned that establishing clear roles for group members to pursue was essential to completing this project on time, and having a reliable medium on which to communicate such as discord was very helpful to making the development of this final project go by smoothly. I also learned that as a team member I should always offer to help my other group members whenever possible, and that is what I did when I let my group members know that they can ask me to explain how pull requests work for this project if they are unsure about that.

Lessons for the Future

From working with Continuous Integration, I learned how different group members can work separately on different parts of a project to create a fully functional project in the end. In the case of this project, I learned how to set up a system that detects linting errors and failing tests in code pushed by group members. This leads me to infer that software projects in the real world may also use similar kinds of automated tests to make sure everyone's code meets a certain standard before being integrated into the main project file. Working with Continuous Integration facilitates better software development because it makes sure that everyone's code meets certain standards as described above and it allows others to give feedback to each other so that everyone can learn different lessons from working on the project.

It is important to have a solid test suite because very often the tests are the only way to verify whether or not the entire code written actually works. If there is a failing test, then it is necessary to first make sure the test was written correctly and then determine if the actual program being tested needs to be modified. Once all tests have been added and there are no more new tests that should written, then the final project can be considered complete. From completing this project, I learned that there should be many tests that try to cover possible edge cases that the original program tested may have missed. The tests shouldn't be completely exhaustive like test one million inputs from 1 to 1,000,000 as doing this would take up too much resources in a real software development project and that there could be more intelligent ways to write a certain test, such as testing with 0, 3600, 86399, 86400, etc. in the context of my project since those numbers represent seconds in an hour or in a day. Therefore, after completing this project, I learned to make sure to communicate with other group members well and positively while never blaming anyone for any problems and to make sure the final testing suite is well-written so that passing all the tests gives all group members a high degree of confidence that the final project satisfies all requirements.