Andrew Sychtysz

Professor Nagpal

Software Process Management CS-348-01

13 November 2020

Advanced Assignment #4

Issue #	Location			Clean Code Chapter # and Practice
	File Name	Line #	Issue Description	
1	GameOfNim.java Project5.java	Many Lines	Most of my last change in the codes have to do with cleaning up the spacing, adding a little more informative comments, and ensuring the code is broken up into sections.	Ch. 4. Informative Comments. Ch. 5 Formatting.

Reflection Report

1. Describe the changes you made and the practices/principles that you applied to make those changes.

- These two codes have undergone a series of changes. During this entire assignment, I have updated two different projects. The second project, “Game of Nim” was a larger code which had undergone many changes. To explain “Roman Numeral Conversion” project briefly, it was a little challenging to “clean up this code”. When I first was learning how to code, I was told about “bad coding habits” immediately. When I first started to learn about coding, people would tell me terrifying stories about their own experiences. Many of my friends that have tried coding outside of the University warned me about how challenging it could be. They would tell me horror stories of “missing a semi-colon” and the entire program not compiling. I soon began to understand that the reason some of my colleagues had such trouble coding was because they were using poor coding practices. I would say, this Roman Numeral Conversion project was one of the first codes I had written where I became confident in my coding abilities. I managed to complete this project on time and ensure it did what it was supposed to do with no errors. When I went back to "clean up" this code, I found that I had some strong coding practices at the time. But there was room for improvement. Not only did I find that there were names of variables and parameters that Uncle Bob would not approve of, I found that I might have even made the project harder on myself with some simple variable names that I came up with. So I changed them. I made the code much cleaner for myself and anyone else who reads Java to understand it.

- Moving forward to my "Game of Nim" code, now this is where the true experience of "cleaner code" shined. Working on this code was a great experience. This code was written after my "Roman Numeral Conversion" code. As I reflect on the past, I recognize that this was a code I went in over my head. I was asked to create the simple game, and I went overboard. Not only did I go overboard and attempt to make more for my program that was asked of me, I wrote these "extra" features before I had even completed what was asked for me. It caused a very hectic coding experience and I struggled to complete the project on time. Although I managed to complete it, the code was not clean at all. I wasn't happy with what I turned in. I ended up having to delete my "Player records" feature when I turned it in. I wasn't satisfied by this, I actually revisited this code one last time after submitted in effort to complete my "Player Records" feature. I never managed to fix it until I went back to it for this "cleaner code" assignment. My code was too messy to make this feature stable. It wasn't working properly due to my static variables being placed in the wrong "if" statements. Once I organized and cleaned my code, it was a very easy task to ensure that the feature would work properly. Now, a year later this "Game of Nim" code finally runs how I envisioned it to. And I am very happy with the result. A lot of passion went into this code from the start, and the final product is beautiful.

2. Explain why you feel that there are no other reasonable places to refactor.

- For the "Roman Numeral Conversion" project, I knew there were no other reasonable places to refactor because even some of my changes did not seem that necessary. The code worked perfectly fine, there were just a few things that made it would make it easier to read and understand. But for the "Game of Nim" project, there was another reason for why I knew there were no more reasonable places to refactor. I enjoy video games and since this project was a game, I knew there was nothing else to fix when I started to think of new features for my game. It got to the point where I could not find anything to fix and wanted to create more things for this game. The code worked exactly how I wanted it too, it looked very clean, and it got to the point where I didn't want to ruin the beauty of the current state it was in. I found a lot of benefit in refactoring the "Game of Nim" code. The only thing I could see Uncle Bob complain about was my "If" statements being so long. Unfortunately, these statements were integral to the structure of the game, and for me to make them one liner would require me to rewrite the entire code with many more methods. These were not simple methods to begin with. I designed the AI players and the human players functionality through a single method. If I were to rewrite the code from scratch, I would have written multiple classes and methods. But I wrote this program in the early stages of my coding days, so I was not capable of these more complex coding practices at the time.

3. Discuss how you feel about your original code, and how you feel about your fully refactored code.

- As I stated before, a lot of passion and pride was put forth into these two projects. I don't think my "Roman Numeral Conversion" needed too much refactoring but it was nice to do so. For my "Game Of Nim" project, that was definitely an amazing time to refactor. I managed to fix one of my features and make the code much more legible. On top of that, it is just so much more beautifully formatted and organized. I think my favorite part about this assignment was taking some of my old projects and throwing it into my own repository that I created. It was very fun have complete creative control over my repository. I enjoyed changing the readme.txt's and looking at all of my commits. The final product looks beautiful to me on Github.