Kopanskaboonify
Sam Kopansky

CS 1632 – FINAL DELIVERABLE

Software quality assurance is something that I know I need to be excellent at if I want to be a developer in modern day America. If I made software that was of horrible quality, didn't match any functionality that it should, or simply didn't work, I wouldn't have a job for very long (assuming I can get one in the first place).

I took this class to become a better programmer. I wanted to know where I was going with my programs and how I would lay them out before starting to code. I have always used the "Just start typing and you'll figure everything out along the way" approach and although it hasn't had particularly bad results so far, I could certainly see many places where I could have improved my code or organized things more efficiently.

Not to toot my own horn here, but because of the project I chose to do, I have created the most incredible program I have ever written. Just looking at it makes me happy. I've never made anything so clean, organized, and functional before in my entire life. Quick disclaimer, this program doesn't cure cancer or anything magical, it's just a silly game that I made up. I'm just really proud of how amazing code looks when using Test Driven Development (TDD).

Test driven development in its most basic form is writing tests, and then writing code that passes those tests. That's exactly what I did. First I wrote tests, which took much longer than I thought since I had to think of how a method could be made to be testable in the first place, and then I wrote code, which took much less time than I thought because I already had an idea of how the code will be written from when I wrote the tests. It was amazing. Every line of code came out and perfectly fit with the next because every line of code did exactly what it needed to do how it needed to do it because of the aforementioned tests.

I think I'll be writing code very differently from now, exclusively using TDD when possible, unless I find some better way to code, but who knows when that might come into my life. As of now, I've fallen in love.

Now, let's take a look at the actual program. It's called Kopanskaboonify. It takes a number that you put in, converts it using a secret equation, and tells you if your number was correct or incorrect for the Kopanskaboonify equation at that moment. Personally, I think this

software could make millions. It's more addictive than Candy Crush and can be played on any command line in the world since it's written in Java. The in-game advertising potential here isn't great, but just by seeing it once, you know it's worth the $999.99 download price, which will make the game pay for itself after the first download. If that's not an incredible ROI, I don't know what is.

It has no errors and all written tests pass. These tests cover 100% of public methods other than the main method. Since I have been having issues with the JUnit plugin in eclipse, I decided to make my own test program that did everything JUnit did to make sure everything passed and was covered. Screenshots can be found below. Once you take a look at the appealing interface and slick game flow, I'm sure you'll agree with me when I say this is definitely ready to release to the public asap. Let me know what you think!

Full project available at: https://github.com/SamKopansky/Kopanskaboonify