Ethan Cheng, Per 2, formerly Per 1.

Our project development was very systematic. Personally, I said that the topic would be decided by whoever my partner was since I was already working on a personal project. I threw in a few suggestions, but he came up with the overall concept, which was a game about playing as the GitHub Octocat and trying to kill all the bugs in the GitHub servers. We had a checklist of things that had to be done, and other things that were planned as extra features. The main development phase with the most work was during the winter break, when Jesse was in London and I was bored at home, so I made the majority of the first version on my own in the span of around 3-5 days. This reflects our division of work pretty well. Usually it was me just coding because I enjoy it and I usually code when I'm bored, and then I tell Jesse was else I need done and he does it. I actually came up with the idea of using git branches to save snapshots with different versions of our project, so the first version is in a branch called Stable-0.01a, where other branches are called Stable-0.02a, Stable-1.00b, etc.

Eventually, we got all the stuff on the checklist done. I am pretty content with this project, but I am not fully satisfied. This project was done completely in Java Swing, since I chose not to use Processing. I learned a lot about graphics processing, as well as multithreading, music playing, and memory management in Java. It was my idea to multithread the entire program to make it faster, but my main disappointment was that I did not synchronize the threads, so there are glitches with things happening to slowly or to quickly. A lot of my learning experience came from ZetCode, which is a great resource for learning the basics of coding in different frameworks.

There were a lot of issues with the game mechanics and balancing the game, because at first people told us that you couldn't pass level 4 (out of 31), and then after a balancing change, levels 1-10 were too easy but the boss level on level 11 was virtually undefeatable, to which I said: “It's not the game, you're just not good enough at it.”

The way sprites were rendered is also very trivial. I was too lazy to programmatically rotate or reflect the pictures, so I had separate images for every orientation of the picture, resulting in 30 MB of extra downloads. I also added background music in the last half hour because I thought the game was too bland. I think I could have done this project in September, especially with some time dedicated to learning Java first.