COMPENG 2SH4 Project – Statement of Contribution

Your Group Number	PPA3++
Your Name	Paul Stoica
Your Team Member's Name	Aidan Harris

You must complete this statement of contribution without discussing it with your project partner, i.e., individually. Your statement should be concise (at most one-and-a-half page). It has three parts:

1. Tell us about your own contribution to the development of your COMPENG 2SH4 project. For example, you can tell us about which project iterations (as mentioned in the project manual) and C++ project classes that you worked on and completed. You can provide a concise answer either in paragraph form or through bullet points.

    - Iterations:
        - 0
        - 1A
        - 1B
        - 2A
        - 2B + food class
        - 3 fts. 1 and 2 (snake body implementation, growth, and food consumption)
    - Other
        - Extensive debugging (mostly of my own work, otherwise bug-free)
        - Made a new method to index through objPosArrayList
        - Made a new class to help print more efficiently
        - Implemented some special foods
        - Extensive commenting, whitespace

2. Repeat Part 1 above but this time tell us about your project partner's contribution to the development of your COMPENG 2SH4 project.

    - Iterations:
        - 3 ft. 3 (snake collision check)
        - Overhauled food class to implement foodBucket and some bonus foods
            - Shorten snake, score +2
            - Extra lives mechanism
    - Other
        - Extensive drmemory support across iterations, as mine was somewhat incompatible with gameplay
        - Brainstormed many ideas for bonus features, many of which were eventually implemented.

The peer evaluation was very mixed in terms of contributions, so it's not worth discussing here.

3. Tell us about your experience in your first collaborated software development through this project – what was working and what wasn't. If you are a one-person team, tell us what you think may work better if you had a second collaborator working with you.

The biggest challenge was that only one of us could make changes at any given time without forcing a merge. As a result, I was somewhat greedy in terms of how much I would work on the project myself, and how long I would go without pushing to the shared repository. This is my explanation for the distribution of tasks observed above. I'll justify my behaviour by mentioning that my partner would often go a long time without responding to messages, resulting in us missing the signup deadline and starting late. When I had to wait for a response before proceeding with something, it contributed to a feeling that collaboration was a rate-limiting step, and that I could produce a better-quality product if I took full charge of the limited time that I had. I will stand by the actions I took based on my judgment, but I will not assert that I accurately judged my partner's capabilities or work ethic.

This approach worked decently well for me as I learned a lot more about OOD than I would have if I did half the work. It's possible that a more active partner would have forced me to pay more attention to encapsulating and modularizing my code to facilitate bottom-up development on their side. However, by implementing a new class I think I did enough of that from my own initiative. I also felt more comfortable towards the end of the project because I knew virtually everything about every class and could debug our code efficiently. The only major weakness was, again, the fact that I gravitated away from collaboration and had to do more work as a result.