COMPENG 2SH4 Project – Statement of Contribution

Your Group	drmemory
Your Name	Jessica Yang
Your Team Member's Name	Maryam Ahmed

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.

    - General debugging, troubleshooting, and ensuring DMA structures are properly implemented in collaboration with project partner when issues arise
    - The player class implementation (iteration 1a), including using FSM to translate keyboard inputs to updates in character position/implement the wraparound and implementing the player movement mechanism.
    - In the project.cpp file, implemented the code for printing out the player character and its movements onto the board.
    - The objPosArrayList Functional Class (iteration 2a) by creating and testing the private data members and member functions in the separate folder before transfer to the main project.
    - objPos class copy assignment operator/function/constructor for proper DMA implementation
    - Implementing the snake body by replacing the data member objPos playerPos with objPosArrayList* playerPosList. Used functions from iteration 2a to create the snake movement. In the drawscreen routine in project.cpp, updated the code to print the entire snake on screen.
    - Implement the routine for checking if self collision with the snake's body occurs, the game over condition, and losing message.
    - Implement spawning of foods from the Food bucket into the drawscreen routine by implementing conditionals to check if a coordinate is occupied by a border, player, and food (in order of priority). Altered conditions for tail unit removal to integrate the increase of player length and movement of player.

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

    - General debugging, troubleshooting, and ensuring DMA structures are properly implemented in collaboration with project partner when issues arise
    - The GameMechs container class (iteration 1b), including the game board size parameters (X and Y), the input collection routine in the main function, exit command, end of game routine and clean up routine, including exiting game message
    - In the project.cpp file, my partner implemented the code for printing out the image of the board on screen

- The random food generation mechanism (iteration 2b) by creating a snake food class and implementing the required data members and member functions
- Coding the routine for printing the single food member on screen in the drawscreen routine and instructions for exit command and WASD keys for movement
- objPos class destructor for proper DMA implementation
- The Snake Food Consumption, Snake Body Growth, and Scoring system - implementing the mechanisms to increase the length of the player snake array when the head of the snake interacts with the code
- Implementing objPosArrayList data member 'Food Bucket" for the above and beyond feature, including generating the algorithm for the random spawning of 5 foods and creating special foods.
- Modified player to check for consumption and then increment scores and increase in player length according to food consumed under the check food consumption function.

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.

I thought working on the project in a pair was a good experience — my partner and I had frequent communication throughout the project so we were always both in the loop/knew what stage the other was at. This was especially helpful in better understanding our own parts and how the different sections will interact. Whenever we ran into challenges such as semantic errors, we collaborated to resolve them together by inspecting different parts of the code, brainstorming possible root causes/solutions and testing. I think it was really helpful to have somebody else to work through problems with together and often we were able to catch errors that the other person did not initially see. As well, by combining our ideas, we were able to create strategies to improve the code's clarity / quality. Using github however was a bit confusing to work out at first - sometimes when we pulled code, duplicates of the same line would show up if the numbering varied between the two codesm and the merge editor was confusing to work;