COMPENG 2SH4 Project – Statement of Contribution

Your Group Name Yash-Shiv

Your Name Shiv Patel

Your Team Member's Name Yash Panchal

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 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.

   For this project, I contributed by:

   a. objPos class development (working on both .h and .cpp files)
   b. Player class implementation (.h & .cpp files)
   c. Food class implementation (.cpp file)
      i. Helping implement random positions for food to spawn
   d. gameMech class implementation (.cpp file)
   e. Debugging problems within code, specifically in Player and Food files
   f. Helping fix and upload code (fixing Git-related issues)
   g. Helping implement the main Project file
      i. UI-related, adding specific messages and implementations for the display
      ii. Helping implement classes from different files
   h. Implementing and verifying all memory allocation/deallocation-related concepts
   i. Commenting and polishing up code
2. Repeat Part 1 above but this time tell us about your project partner's contribution to the development of your COMPENG 2SH4 project.
   For this project, my partner, Yash, contributed by:

   a. Updating and helping fix objPos class
   b. Implementing objPosClassArray (via Test Bench, then in code)
   c. Main project file
      i. UI development
      ii. Specific game conditions (ex// post-game conditions (depending on win, lose, or quit), player speed, overall board implementation)
      iii. Game implementation
   d. Implementing and helping fix gameMech class and files (.h & .cpp)
   e. Fixing Git/GitHub related issues
   f. Ensuring all memory-related tasks were taken care of, and verifying memory leakage reports
   g. Commenting and polishing code
   h. Debugging errors in all classes and code

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.

Although this was not my first time working in a group environment for software development, this was a great experience, one filled with plenty of learning curves! Having two individuals, allowed us to implement code much more easily and approach implementing our ideas in specific ways, and with the help of each other, we were able to feed off each other and bounce back ideas to implement the base of the game. When it came to putting it all together, we were able to go back and forth, as once a person implemented a feature, the other partner would try an add-on to make the UI better or an easier way of implementing that feature (i.e. Yash added game instructions and a message advises the player to not crash with yourself, where I took that feature and added it where if and when the user the reaches the body size where they can crash into themselves, it displays a message until the game is terminated that the snake can now crash into itself, please be careful). To add on, having a partner reduced the workload, as after completing the project, we realized that there was a lot we implemented and that doing this on your own would take plenty of time. It was also interesting to see our code file by file, class by class, come together and then implement it all together to complete and form the game. On the contrary, working with local files amongst two partners and uploading and relying on Github was somewhat of a struggle, hassle, and a new experience to overcome. We faced various challenges, as when one person would implement certain variable names and such, we had the ensure the other was able to copy that and implement the code based on how the previous partner may have defined or constructed certain functions and/or classes. It wasn't a major challenge, but something that took time to get adjusted to. Additionally, when it came to Push/Pull with GitHub, it caused some challenges as the implementation of segments depended on each other and sometimes, if we tried to Push/Pull to override a version, we were unable to do so. Although we found a temporary solution, which was emailing the code/files and copying/pasting from previous versions of our code off GitHub or uploading the files to GitHub, it was unnecessary and made it unorganized. Lastly, if I could change one thing, I'd say to somehow work on an IDE that is live and we could make real-time contributions at the same time. I assume there are ways it could be done on GitHub, but that strategy would reduce or even remove the major barriers we faced throughout the project.