COMPENG 2SH4 Project – Peer Evaluation

Your Team Members Christy Pan & Ryan Li

Team Members Evaluated Ryan & Vraj

Provide your genuine and engineeringly verifiable feedback. Ungrounded claims will lead to deductions.

Part I: OOD Quality

1. **[5 marks]** OOD is about sensible code modularization. Looking at the header files of each object, can you easily interpret the possible behaviours of the objects involved in the program, and how they would interact with each other in the program? Comment on what you have observed, both positive and negative features.

Looking at the header files of each object, we can easily interpret the possible behaviours of the objects because they are well-defined. Each header file represents a class, therefore, making it easier to understand and maintain different components of the program. They have implemented an extra class named 'Food,' it was easy to identity the behaviours of this Class, because the header file was clear and specific. Each object is named appropriately corresponding to their function and interaction throughout the program. Based on the object defined as either void, bool, or int we can interpret how the objects behave, such as its return statement.

2. **[5 marks]** Examine the main logic in the main program loop. Can you easily interpret how the objects interact with each other in the program logic through the code? Comment on what you have observed, both positive and negative features.

   After examining the main logic, we can easily interpret how the different objects interact with each other. Their created food class was implemented well to generate food at random positions. In their DrawScreen process, they have used their Food class objects to correspond with their player objects to check whether they are being displayed in the same X and Y coordinates. Their loops utilized the GameMechanics class objects to determine sizing of the board. In their CleanUp process they have used GameMechanics class objects to determine the game end conditions of the Snake Game. Overall, the objects' interactions were easy to interpret.

3. **[3 marks]** Quickly summarize in point form the pros and cons of the C++ OOD approach in the project versus the C procedural design approach in PPA3.

   - Easier to debug in the Project compared to PPA3 as there was individual testing for each classes, making it more straightforward to identify and fix the bug without affecting the entire program.
   - Procedural design follows a linear execution flow, enabling us to follow the logic of the program more easily in PPA3 compared to the Project

- C++ OOD supports modular design through classes and objects, this promotes a more organized program, as well as a more efficient program as there is a reduction in the amount of repeated code. The code is more organized in the Project as each major component was separated into their own Objects

Part II: Code Quality

1. **[4 marks]** Does the code offer sufficient comments, or deploys sufficient self-documenting coding style, to help you understand the code functionality more efficiently? If any shortcoming is observed, discuss how you would improve it.
   - The code provided sufficient comments, which explained the functions of the implementations thoroughly
   - The comments make the code very self explanatory and easy to understand, which makes it easier to follow through its functionality. There is not much more for them to comment.
   - They have also made comments under each object explaining more actions they needed to implement in order to enhance their code. Example: void movePlayer() under the Player.h header file.

2. **[3 marks]** Does the code follow good indentation, add sensible white spaces, and deploys newline formatting for better readability? If any shortcoming is observed, discuss how you would improve it.
   - The code has good indentation, and allows for us to follow the structure of their program more easily
   - Has a little too much unnecessary white spaces. There are areas with double, sometimes triple white spaces. Besides the white spaces the code is well organized and easy to read.

Part III: Quick Functional Evaluation

1. **[6 marks]** Does the Snake Game offer smooth, bug-free playing experience? Document any buggy features and use your COMPENG 2SH4 programming knowledge to propose the possible root cause and the potential debugging approaches you'd recommend the other team to deploy. (NOT a debugging report, just a technical user feedback)
   - The Snake Game offers a smooth, bug-free playing experience, which achieves all of the Project's goals
   - They have implemented a Food Class which generates, and determines the Food's location, instead of placing it in the GameMechs Class, this can be beneficial as it further organizes the functions that are related to the 'Food Class'
   - There may be slight delays in the Snake movements due to the board size which would cause more MacUILib_printf statements but can be easily resolved

2. **[4 marks]** Does the Snake Game cause memory leak? If yes, provide a digest of the memory profiling report and identify the possible root cause of the memory leakage.
    - The Snake Game does not cause any memory leakage. They have properly deleted the allocated memory heap in the 'CleanUp' function and created the proper destructors. Overall, this Snake Game has great memory control.

Part IV: Your Own Collaboration Experience (Ungraded)

1. 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.

    Christy:
    As my first collaborated software development, I would say that I had a very knowledgeable experience. We were able to revise each other's codes and catch different bugs that the other programmer may not have caught. This led to a more efficient program. Ryan was able to share his knowledge and help me with any confusion I had with the project, as well as explain his logic on some of the implementation of his code. At some points of the project it did get confusing as we had different approaches to tackle problems, however, we were able to communicate and explain our ideas. Ryan was overall a great partner to work with.

    Ryan:
    A collaborated software development does have its pros and cons. This collaboration experience has worked out very well. Although there may have been some inefficient approaches towards solutions, all problems were tackled through both Christy's and my inputs. Christy was an amazing collaborator and had excellent communication skills to help get us through this project.