COMPENG 2SH4 Project – Peer Evaluation [25 Marks]

Your Team Members Ricky Huang Tommy Phan

Team Members Evaluated Devin Gao (gao103) Phillip Lee (leep46)

Provide your genuine and engineeringly verifiable feedback. Ungrounded claims will lead to deductions. Completing the peer code evaluation on time will earn your team a total of **25 marks**. Do not exceed 2 paragraphs per question.

Peer Code Review: OOD Quality

1. **[3 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 observing the code, we have determined that the interaction between the classes and objects can easily be interpreted. The code is easy to understand, as most of the of the functions are simple and straight to the point, meaning no redundancy. Other than, the insertHead and insertTail method, had some redundancy with creating another array list to manipulate, when they could have just manipulated the original array list.

2. **[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.

   Pros:

   - Reduce the number of repeated codes such as creating new functions for each object.
   - Can easily be developed, as in you can implement new tasks, and manage the program easier.
   - The programmer can create a more organized program through C++ OOD in comparison to C procedural design.

   Cons:

   - Memory management is difficult as you are allocating memory dynamically and you need to remember to free those dynamically allocated memory.

Peer Code Review: Code Quality

1. **[3 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 commenting is sufficient, but we also feel that the commenting may be quite redundant or too many comments. Such as the incrementing score, and the getter functions, as it is self-explanatory and obvious in what the function is supposed to do based on its name. However, the comments do provide a good explanation for the functionality of the function or method.

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 formatting of the code provides a good experience for the reader to interpret the code, as it contains good indentation and reasonable spaces between each line of code. Additionally, as a player, they printed out text uses newline formatting which makes the program looks neater. Ultimately, the program is formatted well and contains no formatting errors.

Peer Code Review: Quick Functional Evaluation

1. **[3 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 technical user feedback)

   The game does provide a bug-free gameplay, as it does run intendedly, there are no buggy features within the game. We would say the game is well implemented and bug-free. Conclusively, the player will have a smooth playing experience.

2. **[3 marks]** Does the Snake Game cause memory leak? If yes, provide a digest of the memory profiling report and identify the possible root cause(s) of the memory leakage.

   After running DrMemory, the report provided 0 leaks and no potential leakage of memory. Therefore, we can conclude that the program does provide a good experience for both the user and the system it is played on.

Project Reflection

Recall the unusual objPos class design with the additional Pos struct. After reviewing the other team's implementation in addition to your own, reflect on the following questions:

1. **[3 marks]** Do you think the compound object design of objPos class is sensible? Why or why not?

   The compound object design of the objPos class is sensible as it provides a foundation to create further objects with a variety of different coordinates within a given parameter. Without this, it would be very difficult to implement the ObjPosArrayList class which is used to create the snake body and store the food objects. The objPos class is also required to for the implementation of collision detection.

2. **[4 marks]** If yes, discuss about an alternative objPos class design that you believe is relatively counterintuitive than the one in this project. If not, explain how you'd improve the object design. <u>You are expected to facilitate the discussion with UML diagram(s).</u>

In this new objPos class design, we have declared x and y as a char, while the symbol will be initialized as ASCII value of the char we want to represent as the symbol. We have included 3 extra getter method, the change the value of x and y into an integer that represents a coordinate in each parameter and saved as xInt and yInt respectively. While the last method will use the ASCII value of the symbol to change into a char with that ASCII and save it into another variable of symbol. Now instead of using a struct, and creating new Pos, we can just simply equate the xPos, yPos and sym, to the above variables. Below is the UML diagram: