

COMPENG 2SH4 Project – Peer Evaluation [25 Marks]

Your Team Members Gokul Radheepan

Team Members Evaluated kirithd (400517864), diasj11(400541065)

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.
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.

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.
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.

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)
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.

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?
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. You are expected to facilitate the discussion with UML diagram(s).

OOD Quality:

1.

The quality of the code is excellent, and very easy to follow. Additionally, their main is very modular and simplistic, with adequate comments to describe the logic

2.

The benefits of OOD include:

- Modularity and encapsulation, with code into classes making it easier to understand the flow of code
- Reusability and Maintainability, with objects being able to be instantiated and reused, as well as maintainability making code isolate and more testable

Cons:

- Complexity, especially when planning the structure of code
- Over engineering, where OOD could be used in cases that does not require it, resulting in a more complex solution

Code Quality:

The code has reasonable comments in areas that require explanation, and line and indentation spacing is spot on for great legibility. Explanation of code in comments is done concisely and not overdone, making the readability very easy. Normal conventions were followed for indentation and line spacing. Additionally, the game screen is outputted very nicely and has all the required information on display and functional.

Quick Function Evaluation:

1.

The code works fine with no apparent errors. The snake grows, wraps around, and dies when hit by itself. The code works bug free with nothing else to improve on.

2.

There is no memory leak, and all objects that require a rule of 4 have it implemented, as well as destructors in all object classes that require it.

Project Reflection:

1.

In my opinion, having the struct within the class does not make much sense to me, as having x and y be private members of the objPos class should be encapsulated enough for the sake of the project. Another design change that could also be implemented would be to make the symbol attribute private as well, but overall would not change the actual

2.

To improve the design, I would get rid of the struct and make x and y attributes of the objPos class and leave everything else the same. There are a few getters and setters that could be added as well, for instance in my code I had getters and setters for the specific x and y positions of the object, but those were overall unnecessary for the overall usability of the class. Additionally, the symbol attribute of the class could also be encapsulated.

