COMPENG 2SH4 Project – Peer Evaluation [25 Marks]

Your Team Members _____zhuk50_____ _____Khans294 _____

Team Members Evaluated ___kukrejap_____ _____virka12_____

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.

   The objects use clear variable names to show how they interact with each other, and how the program logic affects their generation and movement in the main program loop. A few sections, such as in RunLogic, unnecessarily increase the length of the code and adding more logic to be aware of, as some of the logic would make more sense in another file. However, this does not affect the clarity of the overall main program loop.

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.

   C++ OOD Pros:
   - More organized, easier to keep track of all objects and functions
   - A more intuitive logic after beginning

   C++ OOD Cons:

   - Switching back and forth between files for function names and the order in which multiple files must be called to access something in the last one

   C Procedural Design Pros:

   - One logic flow, easier to start the code
   - Going from one part of the logic directly into the next; all code is in one location

   C Procedural Design Cons:

   - For longer games and bigger projects – less organized
   - Must use either global variables or call the variables inside of every function, less organized

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.

   Overall, the code offers sufficient comments for having a better understanding what each part does what within the project so any user or reviewer can understand the purpose for each classes or functions that are called and this makes it easier for us by removing any confusion.

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 follows good indentation as the code has spaces and new line where it feels easier to read and understand and it's not like cramped together which is great for readability and this is a good programming practice that is vital for both the programmer and reviewer.

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 runs smoothly and does not run into any noticeable bugs.

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.

   The Snake Game does not cause any memory leaks.

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?

   I don't think the compound object design of objPos class is sensible as it leads to memory leaks if you are not careful and complexity due to copy construction and assignment.

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).

   In order to improve the design of objPos class it's better to include it as a direct member of the objPos class instead of dynamically allocating the Pos struct and this will help with memory management and efficiency.