COMPENG 2SH4 Project – Peer Evaluation [30 Marks]

Your Team Members	Omar Bayari	Nabil Memon
Team Members Evaluated	Hannah	Julie

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 **30 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 main program logic looks very smooth and easy to read. All functions were in the correct place, whether in RunLogic, DrawScreen, or GetInput. However, there were some global variables incorporated in the Project.cpp file. These should have been part of the gameMechs class and not global variables.

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 of C++ OOD approach:

- Classes and objects make a more organized code structure.
- Code is more readable and enables code reuse.
- Easier to update and manage while working on the code
- Makes the code easier to scale without having to delete a lot of functions and modify them

Cons of C++ OOD approach:

- More complex and uses more advanced concepts
- Risk of making simple solutions unnecessarily complex
- Can make tracing errors harder; a lot of different classes to keep track of

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.

   Generally speaking their comments are sufficient enough, telling me what each line of code does if it isn’t already implied or simple enough. Some comments however, should have been removed as they were used to while making their code (thoughts and opinions). Those comments can be slightly confusing as they aren't meant to be read by any other coders. There are also a few lines of code that were not needed that were commented out that should have been removed and not commented out.

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 excellent indentation, white spaces and new lining. The code is easily readable to any coder.

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 is mostly smooth overall and there are no major bugs. However, the snake does not immediately grow when food is eaten. It takes a couple iterations for the snake to grow. This is due to an incorrect growing algorithm. The possible root cause is in the objArrayList functions, specifically in the insertHead function. I would recommend using the debugger and analyzing where the program goes wrong. The snake only grows when the end of the snake reaches where the food originally was. The snake should grow as soon as the food is hit, and the head should be inserted.

   The game also changes the position of the food when the character 'F' is pressed. This is an unnecessary feature.

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.

   There is no memory leak.

Project Reflection

Recall the unusual objPos class design with the additional Pos struct. After reviewing the other team's implementation in addition to yours, 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 objPos class which has includes nested Pos struct, is sensible when it comes to much larger projects however in this case it would be unnecessary as it makes it slightly more complicated.

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.

Instead of having the Pos struct, we can have the objPos include the x and y coordinate in the class itself under the private members and have getter and setter methods in the class as well. This way there will be no need to include the struct.

ObjPos
- x: int - y: int - …*Rest of the ObjPos private members as it was before ….*
+ getX(): int const + getY(): int const + setX(): int + setY(): int ….*Rest of the ObjPos public members as it was before….*