**COMPENG 2SH4 Project – Peer Evaluation [30 Marks]**

Your Team Members John Bacopulos (400505130) & Arther Assaf (400522187)

Team Members Evaluated Milana Kalinic (400503797) & Noor Azam (400503826)

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.
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 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?
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.

**Peer Code Review: OOD Quality**

1.  
Positives:

* Very Object-Oriented programming style, most of the main program logic uses interactions between objects to fulfill the game needs.
* Easy to follow the flow of program execution, as well as the task each function performs

Negatives:

* Food could be improved by creating an Object to hold it, instead of having all of the functions in GameMechs.
* Global exitFlag is redundant as the main project loop should use the exitFlag from gameMechs.

2.

Pros

* Components are reusable as they are stored in classes / objects
* Main project file is cleaner and abstracted so that implementation is hidden
* Easier to maintain / expand functionality as changes can usually be localized to a class or 2.

Cons

* Can be hard to keep track of memory allocation and deallocation
* For smaller projects, might not be worth the extra complexity

**Peer Code Review: Code Quality**

A screenshot of a computer program

Description automatically generated1.

The code has sufficient comments and is very easy to follow.

For example, the commenting in the DrawScreen function help to convey what each loop does.

2.

The code display sufficient indenting and whitespace. Nested loops are correctly indented and therefore easier to follow. Only shortcoming is lack of consistency, for example sometimes objects have functions being accesses by “gameMechs -> getFoodPos()” and other times there is no space between the arrow.

**Peer Code Review: Quick Functional Evaluation**

1.

The program runs smoothly with no bugs. Very responsive. Game ends immediately on death condition.

2.

A computer code with white text

Description automatically generated with medium confidence  
Seems to be one memory leak related to the allocation of gameMechs at line 23 in Project.cpp.

**Project Reflection**

1.

I think that the Pos struct adds unnecessary complexity to the objPos class design. It would be much easier and simpler for the objPos to just have and x and y private variables. It also removes the need to allocate memory for the Pos struct.

2.

Like mentioned above I would improve the objPos class by just having x and y be private variables, with getter and setter methods. The UML diagram would look like:

A screenshot of a computer code

Description automatically generated