

COMPENG 2SH4 Project – Peer Evaluation

Your Team Members Logan Eby

Team Members Evaluated Anusha and Jenisha

Provide your genuine and engineeringly verifiable feedback. Ungrounded claims will lead to deductions.

Part I: OOD Quality

1. **[6 marks]** OOD is about sensible code modularization. Looking at the header files of each object, can you easily interpret the possible behaviours of the objects involved in the program, and how they would interact with each other in the program? Comment on what you have observed, both positive and negative features.

Just by looking at the header files of each object, I was able to interpret the possible behaviours of the objects involved in the program, and how they would interact with each other in the program. I would be able to better interpret this is comments were provided within all header files. Functions and private variables were named properly, allowing me to understand more easily what they did.

2. **[6 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.

Yes, I can easily interpret how the objects interact with each other in the program logic through the code. Objects take up most of the processing, leaving the main code very modular and legible.

3. **[5 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 and Cons of OOD	
Pros	Cons
Code is more organized	Code is more difficult to implement
Code is more scalable	Code can be inefficient
Code can be developed in parallel	

Part II: Code Quality

1. **[5 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.

Code offers plenty of comments explaining in detail how things work and is self-documenting. Some comments could be removed as they are more like developer's notes than comments to explain the code.

2. **[4 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.

Code is inconsistent with indentation. In certain cases indents will be used twice when they shouldn't be, or spaces will be left after an indent making it more difficult to keep track of code. Code white spaces and newline formatting are implemented appropriately. To improve indentation, double check code to ensure that you aren't leaving spaces before loop declarations and that you aren't inputting too many indents.

Part III: Quick Functional Evaluation

1. **[8 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 a technical user feedback)

The gameplay is almost entirely smooth and bug-free. The only issue I could find was that there is no way to exit the game, whether by losing or by pressing a button. To solve this, fix the function `setExitTrue` so that it sets `exitFlag = true` instead of setting `exitFlag = false`.

2. **[6 marks]** Does the Snake Game cause memory leak? If yes, provide a digest of the memory profiling report and identify the possible root cause of the memory leakage.

NOTE: In order to test this, I had to change the function `setExitTrue` so that the code could run to completion.

As per DrMemory, the game causes no memory leak.

Part IV: Your Own Collaboration Experience (Ungraded)

1. Tell us about your experience in your first collaborated software development through this project – what was working and what wasn't. If you are a one-person team, tell us what you think may work better if you had a second collaborator working with you.

Unfortunately, my first collaborated software development did not work. I was left with a teammate who was not knowledgeable or able to communicate. As a one-person team, I think that there were a few sections where a second collaborator would greatly help. First, it would have helped for managing the workload, especially on iterations where you could do portions of code in parallel with one another. This would have made these sections theoretically get done twice as fast. Second, it would have helped in spots where I didn't know how to progress or where I had issues in my code. Having a second person to look at it would have allowed me to troubleshoot the issue more easily.