# COMPENG 2SH4 Project – Peer Evaluation

Your Team Members Yiheng Hu, Steven Zhang

Team Members Evaluated Anushka Menon and Sanya Sharma

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.
   **By looking at the header files of each object, I can easily interpret the behaviours of the objects due to well written comments and easy to understand variable names. Other positives include well structed and grouped up code making it look neat and easy to read. This helps determine what each object does and allows for me to determine the relationships between the objects. Regarding the OOD aspect of code modularization I do not see any negative features.**

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.
   **Just like the header files, the main logic within the main program loop and the interactions between the objects can be interpreted and understood easily. The positives again include well structed commented code, where each section can be told apart with ease and without confusion. Descriptive comments allow for code to be explained without much thinking and overall, the format makes examining the code without difficulty. The only slight negative is that sometimes the comments are above while others are beside the line of code. I think if you keep everything the same (all comments are beside the line or above) it will make the code look even more organized, however that is only a minor issue. Everything else in terms of OOD quality is very good.**

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.
   **The cons of OOD:**
   - Can generate a lot of unnecessary code.
   - Could Cause duplication.
   - Takes longer than the procedural approach. (Procedural can create programs without creating classes or objects)
   - Harder to learn and use than procedural design approach.

   **The pros of OOD:**
   - Group development

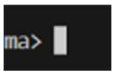
    - People can work independent of each other.
    - Each person can work on different objects at a time.
    - Can break the program into smaller sized parts to solve issues. (easier to debug)
- Reusable modular classes (Procedural can't reuse their code)
- Easy to maintain code.
- Communication between objects is simpler.

## 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.
   **The code offers very well documented comments. Everything from the variables to the logic within the code is well explain with comments. By reading the comments I can understand their respective sections of code easily, therefore in terms of comments there are no shortcomings.**

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.
   **Just like the comments, the structure of the code is well organized and indented correctly. The structure of the code allows for ease on the eyes and grouping of contents provide an easy way to access specific sections (for example if I wanted to see the variables initialization, they are all together). I think there are no shortcomings in terms of indentation and formatting.**

## 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 of the snake game is smooth. Sometimes my WASD keys are not responsive and have to be pressed twice (could be my keyboard or the actual game). The board also has a weird, printed block glitching around that flashes the screen. This puts a slight strain on the eye. For the WASD bug, if it is not my keyboard, checking the switch function and seeing if there's any conflicts could help. For the flashing block issue, try isolating the code where it's printing the board and finding the issue there is probably the easiest. The issue I think with this problem is that it is printing the** 

**part of the terminal. Other than that, please add**

**the exit-flag key for the user in the interface so they know what button to press to quit.**

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.
   **Memory leak is unknown as the project-code causes Dr. Memory to internally crash. This is probably due to errors and wrong implementations of the memory allocation within the code.**

## 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.
   **Coding in a group is much more fun than coding single. With a partner debugging is much easier as you have another perspective on where an error could be. Since the project is done in OOD, splitting up the objects also removes some of the strain. In this project, my group ran into an error with printing the snake in iteration 3 last step where both me and my partner did not know how to fix. After some help from a help session, we managed to fix the code. Other than that one problem, our group did not run into any other issues.**