COMPENG 2SH4 Project – Peer Evaluation

Your Team Members: David Pires and Talha Siddique

Team Members Evaluated: Nathan and Hunter

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.

Looking at the header files of each object and their overall object oriented design, we can see many positives and negatives. Firstly, a positive of the code is that they named their functions well so you could easily interpret what they are doing and what role they fulfill, thus making the code sensible. Furthermore, they correctly determined which parts of their code should be public members and which shoulf be private members such as with regards to the ExitFlag and LooseFlag. Negatives with regards to their code, is that they included class members that they did not need. An example of this would be including "objPos foodPos" as a private member in both the Food and GameMechs class when I did not need this. Another negative is that they failed to include a self collision check function which makes their code more procedural and less object oriented. Overall there are both postives and negatives.

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.

In their main program loop there are positives and negatives. A negative would be that they did not include anything in the Getinput() function of the main logic. This is an issue because it becomes difficult to interpret what they are doing and how their objects are interacting with each other. A positive would be that the initialize() function was easy to understand how the objects are initialized. The runlogic() function was also incredibly simple, and consisted of 3 steps. In the cleanup() function, the way the lose flag was incorporated was very simple and was determined by one if statement:

if (myGM -> getLoseFlagStatus()).

Overall their code is easy to interpret but not completely clear.

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:

- **Modularization:** it is the ability to split up code into different parts, this allows for greater organization as it allows you to sperate different tasks into different .cpp files.

- **Collaboration:** different coders can work on different objects and therefore it makes it easier to split up tasks.
- **Isolation** is another benefit of object oriented design. Changes to one section of code is very unlikely to effect another part of the code due to everything being separated.
- A 4th benefit of OOD would be **reusability**. Objects and be reused through inheritance or simply copied into a different coding project. The same cannot be said for a procedural program which is specific to it's use and is built in to it's program.

Cons:

- Need a thorough understanding of the way C++ classes and objects are designed.
- Requires much more planning and code layout beforehand.
- Everything must be treated as objects.
- Having sperate files containing different object classes and functions may make the code hard to follow.

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.

A negative of the code that we are reviewing is that there are no comments which makes it difficult to determine the thought process of the coder when designing the header files. You could improve this by including a comment for every function to thoroughly describe what is going on in it.

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.

Yes, the code follows all indentation rules and deploys new line as necessary. There are no blocks of text and generally there is a space between every 2-3 lines of code. This makes the code very readable.

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)

There are no visual bugs during the game, and the game terminates correctly and displays the correct message and score at the end.

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.

No the game does not cause any memory leakage. There is 0 bytes of leakage.

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.

Talha:

The things that worked was:

- Scheduling meetings for project development
- Splitting up tasks based on the project iteration.

The things that didn't work:

- Having a thorough understanding of the way the entire project works together, since some members has specific tasks.
- Having to keep up with each other was a bit difficult
- Sharing code with each other was a bit inefficient and confusing. Ensuring both members had the same copy was crucial.

David:

We did a good job of organizing ourselves and getting work done. The instructions along with the tutorial videos posted were very helpful with regards to understanding what we were supposed to do. The instructions were a bit misleading though as the iteration when they said that they were parallel, but we still needed to complete the other iteration to determine if the code was working. Furthermore, GitHub when two users were pushing and pulling was very problematic and it made sharing progress very difficult.