COMPENG 2SH4 Project – Peer Evaluation

Your Team Members	Youssef El Ashmawy	Zahra Kazmi
Team Members Evaluated	Imad	Gurmher

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.

Upon looking at the code, the header files make it easy to interpret what the functions of the files are supposed to do. For example, the gamemechs file holds the mechanics of the game, some positive features are that the function names are clear and concise as they explain what the code is supposed to perform. One con could be that they barely commented on their code which makes it a bit harder to understand the code.

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.

Overall, the main logic in the main loop is easily interpretable. As I am going through the code, I am able to understand why everything is put in each function. The drawScreen function well depicts the grid being drawn on the terminal screen. OOD was also well utilized in the main program loop with accessing the different classes in the project file. Some things to improve would be that the variable names are a bit vague. Variable names like p1 and f1 make it harder to interpret what the variable is holding. Another thing to improve on is to also increase on commenting throughout the development process.

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 :

- Makes it a lot easier to grow the code and add more features.
- Makes the code more reusable with the different setters and getters of each function.
- OOD also adds a lot more structure in the code and makes the code somewhat easier to read.

Cons :

- OOD is very complex to learn.
- More difficult to test than procedural design.

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 has some comments. I do see shortcomings in the commenting in some of the bigger nested for loops where it may be beneficial to the user to explain how those loops work. It also may be beneficial to explain some of the things in the code such as where the game end condition occurs so it is easier for someone to work on the code later.

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.

Indentation is good overall for the loops and conditional statements. It is easy to read the loops and conditional statements. In some of the declarations of loops like shown below, some spacing would be beneficial to take away from the clustered look of the statement.

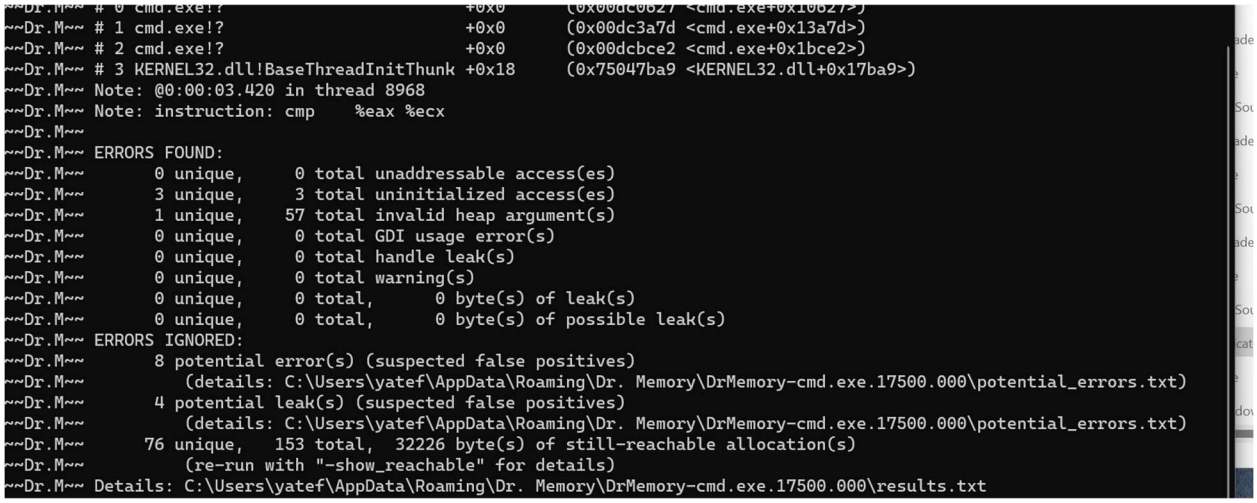
```
for (int i=0;i<2;i++){
                    f1->getFoodPos(*temp2,i);
```

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 code overall runs smooth. There are two very noticeable bugs. The first is that the game will randomly end sometimes and say that the snake touched itself but that was not the case. This is an issue most likely from you raising the exit flag when the game is not supposed to end. The next is the game over text comes too late to the user. This is because of the placement of the statement.

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 memory leakage was found in the code