**COMPENG 2SH4 Project – Peer Evaluation**

Your Team Members Sydney Sochaj

Team Members Evaluated Belal Armanazi and Aarib Syed

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.

6/6

Examining the header files for each object, it is evident that the correct naming conventions have been employed. This makes it straightforward to discern the intended functionality of each function within the objects. The naming conventions, along with associated variable names, facilitate easy identification of functions that may collaborate. The organization of header files is well-structured, with appropriate indentation and spacing. This contributes to the clarity of different sections within the functions, aiding in the understanding of which functions perform similar tasks and how they might interact. The code exhibits positive features in terms of OOD, with a clear emphasis on readability and logical organization.

1. **[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.

6/6

The main program loop is clear and easy to follow. It's simple to see how the functions from other files work together in this main Project file. The file is well-organized, with good indentation, making it easy to understand. The comments in the code are helpful. They're straightforward yet provide a good description of what each part is doing. What's great is that the comments aren't repeating information, and they're easy to grasp. I particularly appreciate how the program shows the player's direction and location during the game, adding a nice touch to the gaming experience. One standout feature is that, when the user hits the exit flag, a new screen appears, showing the score and indicating the end of the game. These additional features contribute to making the game user-friendly and enjoyable. Overall, positive points for clarity, organization, and user engagement.

1. **[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.

5/5

*Pros of C++ OOD Approach:*

* Code is divided into separate files for each object, making it easier to manage and understand.
* Objects, like the player class, are reused in different parts of the program, saving time, and reducing redundancy.
* The reduced amount of code enhances efficiency and lowers the chance of errors, simplifying the debugging process.

*Cons of C++ OOD Approach:*

* Constantly searching through different files to understand functions and interactions can be challenging.
* Understanding interactions between different classes may be difficult initially.
* Debugging can be challenging as errors in functions from other classes might be harder to trace.

*Pros of C Procedural Design Approach (PPA3):*

* Having all functions in one file simplifies the remembering of their purposes.
* Procedural coding allows for clearer visibility of what functions are doing and how they are used.

*Cons of C Procedural Design Approach (PPA3):*

* Lack of modularity makes it harder to manage and scale the project as it grows.
* Code reuse is more challenging, potentially leading to redundancy.
* The larger codebase may increase the chance of errors, making debugging more complex.

**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.

5/5

The code effectively uses comments to enhance understanding. These comments strike a good balance; they are neither excessive nor redundant, presenting a clear and concise explanation of the code's functionality. The comments wisely avoid unnecessary details and refrain from over-explaining the reasoning behind specific functions. This approach, steering clear of intricate technical explanations, contributes to the overall clarity of the code.

1. **[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.

4/4

The code adheres to effective formatting practices, enhancing its overall readability. The implementation of clear indentation and the thoughtful use of spaces within if-statements contribute to a code structure that is easy to navigate and comprehend. The strategic placement of newline characters, especially in conveying messages, adds to the user-friendly experience of the game.

A screen shot of a computer

Description automatically generated

Additionally, the presence of spaces between different sections of the code serves to delineate distinct actions, aiding in the understanding of the program's flow. It's worth noting that these positive formatting aspects are consistently evident across all files.

**Part III: Quick Functional Evaluation**

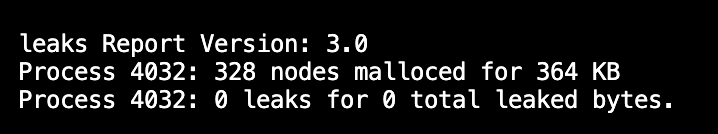
1. **[8 marks]** Does the Snake Game offer smooth, bug-free playing experience? Document any buggy features and use your COMPENG 2SH4 programming knowledge to propose the possible root cause and the potential debugging approaches you’d recommend the other team to deploy. (NOT a debugging report, just a technical user feedback)

8/8

The Snake Game delivers a seamless and flawless playing experience. Key functionalities, including the correct wraparound of player objects along the border, accurate food generation, proper game termination, and precise movement of the snake's body, all contribute to a satisfying gameplay. A minor visual quirk is observed during the border wraparound, where the player momentarily seems to touch the border. However, upon closer inspection of the code, it becomes evident that this is a perceptual illusion caused by the swift movement of the player. It is noteworthy that the development team has demonstrated a high level of proficiency, anticipated potential issues, and proactively addressed them. This attests to a well-crafted project with comprehensive error prevention measures in place.

1. **[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.

6/6



There are no leaks present in this program. The allocated memory is effectively managed and released as needed during the execution of the game.

**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.

If I had another person working with me on this project, it could have been more helpful. When I got stuck on a part of the code, having someone else to discuss ideas with could have made things easier. Getting a second perspective might have helped avoid mistakes that I couldn't easily spot on my own. It would also save time – when I had an error, I could ask someone to check what I was doing wrong, saving time on troubleshooting. Additionally, having a partner could bring in fresh ideas for developing new features, making the project more interesting and innovative.