COMPENG 2SH4 Project – Peer Evaluation [25 Marks]

Your Team Members Marven Denha Matteus Jedryk

Team Members Evaluated Margaret Bui Zahra Kazmi

Provide your genuine and engineeringly verifiable feedback. Ungrounded claims will lead to deductions. Completing the peer code evaluation on time will earn your team a total of **25 marks**. Do not exceed 2 paragraphs per question.

Peer Code Review: OOD Quality

1. **[3 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.
   The main project file/code is written concisely and straightforward which facilitates ease of understanding. Variables are initialized in a clean manner and in their appropriate functions which makes it easy to link variables to their respective objects. The formatting of the DrawScreen function is easily understandable due to the implementation of indenting and appropriate commenting where necessary.

2. **[3 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	Cons
C++	• Control memory usage more efficiently • The use of classes makes the code cleaner as there are less functions on a given file • Due to the use of multiple files we can isolate and debug errors more efficiently	• More memory leakage if not delt with appropriately • Generally harder to use due to having multiple classes that are linked to eachother
C	• Using a single file as opposed to multiple files makes referencing functions and objects easier • Memory debugging is easier to use	• Working in one file is less organized and generally leads to larger amount of lines • Repetitive due to having to call functions multiple times

Peer Code Review: Code Quality

1. **[3 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 does contain sufficient commenting which makes it easier for any who's not familiar with the code to understand. For example, in their Move Player function, they did an amazing job commenting on how the collision process worked. One note to add, unlike our team, they chose to go with the class method for the food generating mechanism, which could have used just a tad more commenting since it is a new class being added. All in all, they effectively commented all their major functions which could be hard to understand.

2. **[3 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.
   Once again, they exhibited excellent use of indentation and formatting of their code. Code is written it its expected and appropriate sections. For example, the Draw Screen function, which is one of the most complex/lengthy functions to write in the project, was indented in a clean and concise way making it extremely east to understand. This can also be observed with their food generation and move player functions.

Peer Code Review: Quick Functional Evaluation

1. **[3 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 technical user feedback)
   When playing the game, there were no observed bugs, and it functioned as intended allowing for a smooth playing experience doing all the required movements and features. One thing to add to make gameplay more user friendly, would be to add more instructions on the game such as the exit command as it is not provided for the player and could be any key.

2. **[3 marks]** Does the Snake Game cause memory leak? If yes, provide a digest of the memory profiling report and identify the possible root cause(s) of the memory leakage.
   No, after running the game and drmemory, we observed 0 leakage and 0 possible leakage, meaning that their deallocation of memory, which was observed on their code, had the necessary functionality to properly deallocate the memory heaps used.

Project Reflection

Recall the unusual objPos class design with the additional Pos struct. After reviewing the other team's implementation in addition to your own, reflect on the following questions:

1. **[3 marks]** Do you think the compound object design of objPos class is sensible? Why or why not?
   No, the objPos class has a pointer to Pos struct, which has X and Y coordinates and the symbol, looking at it from the programing side, it would make it more complex to have pos as a pointer, because of having to add a destructor to deallocate Pos and implement the rule of the 6. Since Pos is a not a huge struct, it wouldn't make sense to have it as a pointer and add more complexity to it.

2. **[4 marks]** If yes, discuss about an alternative objPos class design that you believe is relatively counterintuitive than the one in this project. If not, explain how you’d improve the object design. You are expected to facilitate the discussion with UML diagram(s).

objPos
-Xpos : int -Ypos : int -Symbol : char
+objPos() +ObjPos(xPos : int,yPos : int, sym : char) +setObjPos(objPos o) void +setObjPos(xPos : int, int yPos : int, sym : char) void +getObjPos() : objPos const +getSymbol() : char const +isPosEqual(const objPos* refPos) : bool const +getSymbolIfPosEqual(const objPos* refPos) : char const

Since Pos isn’t a pointer struct, we can now write the class without including a destructor, copy assignment, and a constructor in the class which would make it less lines for the code to run and not only that, but it would make it less complex to write the program, also it would be easier to reference any of the objects including Pos. Not only that but the setObjPos method will help update all the values for the x y and the symbol if needed. So this UML diagram and plan optimizes the efficiency of the code and the readability of it and understanding.