COMPENG 2SH4 Project – Peer Evaluation [25 Marks]

Your Team Members	Jerry Wu (wu744) & Heidi Hui (huiy13) hedden3rd-floorsouth
Team Members Evaluated	basiounr & mohas40 (macID) reesh

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.

   For the most part, the main program behaves like a normal object-oriented program with many different classes and the interaction between the objects is relatively clear and well explained. The main program does have some unusual features though.

   To start, there are three global variables defined at the beginning of the program, which negates one of the key advantages of OOD. The index integers can simply be initialized as local variables wherever they're needed, and as for the input character and the time delay integer, both can be stored as an attribute in the GameMechs class and called in relevant function instead. There is a significant amount of logic in the drawScreen function in the main program, which leaves the function unnecessarily cluttered. Instead the generation of the array containing the necessary information for the game board should be done before the main logic, and the modifications for cells occupied by the snake as well as by the randomly generated food should be inserted into the array by calling the setter method with the appropriated arguments either in runLogic or in the Player and Food classes.

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.

   Object oriented programming offers a modular structure to a program, where functions are defined within each class and include all necessary input (setter) and output (getter) methods required. This means that:

   - The main program only includes essential pieces of the execution of the code and the large-scale actions that need to be done
   - Detailed logic and operations are isolated and specific to their class, only accessing other classes/objects and their functions when necessary
   - Complex objects have a systematic way to be set up within the appropriate class, so multiple instances of the same type of object won't need to be set up manually each time
   - All information passed between objects and classes are done so using pass-by-reference and pass-by-pointer, so no global variables are required

   While the modular approach means each class is simpler than the combined procedural equivalent, it can also lead to more difficulty debugging and modifying key components of the program, where multiple classes need to interact with each other. This approach requires the careful monitoring of where classes, objects, and methods are called in each part of the program.

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.

   Commenting is provided for portions of the code that have a unique design and/or have a more complex logic, including the bonus proportion and the unique customizable speed design. More comments on how objects and methods interacted with each other could've provided more insight on how the program as a whole works, but overall the functionality is clear provided the reader has a good understanding of this style of OOD.

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.

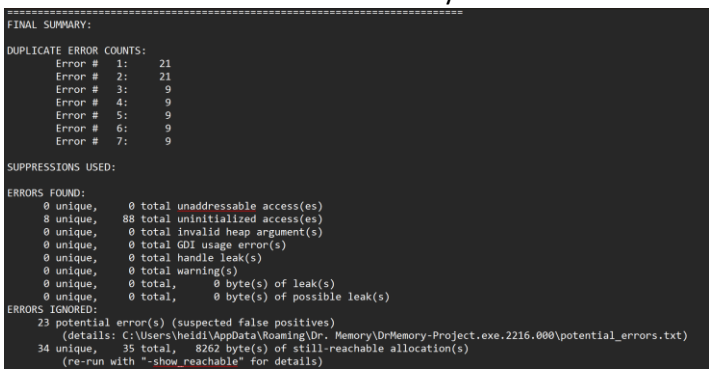
   The usage of indentation, spaces, and newlines are appropriate and make the code legible and easier to follow, and is consistent throughout the program with exception of a few small parts where there are half indentations and extra spaces. These exceptions aren't significant and don't affect the reader's understanding of the program.

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)

   The snake game provides a smooth, bug-free playing experience. They included the changing speed function to cater the needs of the player. It might be a little bit slow and looks laggy when it is at the very slow mode, but it is fine as the delay time of each loop is longer and cause the overall speed to be slower. This is nothing related to a bug or any weird behaviour from any part of the code.
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.

   There are no memory leak in the snake game. Although there are 88 uninitialized access shown and 8262 Bytes of still reachable memory, as there is 0 possible leak and leak in the result of drmemory, we can conclude that there is no memory leak .

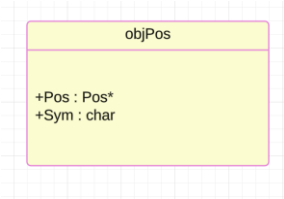


Project Reflection

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?
   I don't think the object design objPos class is sensible. The major reason is the typedef is unnecessary and did not facilitate the overall design. First, typedef is more a C design than C++. Also, the x and y can just be the component of the objPos and it will work perfectly fine and easier to use. However, with the specific Pos member, it makes that every time we access the x or y coordinate and extra step to reach its data. In conclusion, it is not sensible as it is very unnecessary to have the Pos member.
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).
   I will remove the typedef part and the member Pos in the class. Instead, I will add a private member x and y which the variable type is integer. I will also change the symbol to a private member. In order to allow other function to access the private member, I will add in total of 4 get and setter method. For example, getX() const, getY() const,getSym() const, setX(int new_x), setY(int new_y), setSym(char new sym). These method will be placed in the public session of the code to allow other class to use it.

   Old:



   New: