COMPENG 2SH4 Project – Peer Evaluation

Your Team Members Aliasgar Ukani Majed Kalaoun

Team Members Evaluated Samer Vender

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.

Yes, the header file names do reflect the behaviours of the objects involved in the program. For example, the header file "Food.h" implies that the class will contain data members and methods concerned with the generation of food on the board, and that is indeed the case. Another example is the header file "objPosArrayList.h." The name implies that the class will be concerned with the generation of array lists, and that is indeed the case.

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.

Yes, you can easily see how the objects interact with each other in the program logic through the code. This can be seen from the very start of the code where the object pointers are initialized as global variables, which are then initiated as heap members in the initialization stage of the code. Throughout the program you can see objects being declared, in which its member functions are also being used in order to obtain specific attributes needed to run the logic. This can be seen in all the functions, but a great example is in the "DrawScreen" function. In this function, there are objects being declared and used to print out the necessary contents on the screen. Negative features include declaring object variables and pointers within the functions even though they could have been declared in the global variable scope. This would make the functions look cleaner and more concise, as well as increase execution efficiency. A great example is in the "DrawScreen" function where two object pointers and two object variables are declared. These pointers and variables could have been declared at the top in the global scope and could have been assigned within the functions. This would have made the function smaller, thus making it more concise. It also would have made the code more organized as well as more efficient, as there would not need to be new memory declarations every time the "DrawScreen" function is called.

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

- Having multiple classes structures the code in a way such that multiple people can work on it with ease. This is because each person can be assigned one or more classes without necessarily having to be familiar with other parts of the program.
- The main program looks much cleaner. This is because instantiating one class in main performs the work of dozens of lines of code.
- It is easier to see how everything interacts with one another, since each class has a name that reflects its specific role.
- Creating a class allows multiple instantiations of said class with ease, since each class functions as a data type. Generating a new instance of a class is as simple as declaring a new variable with said class as the data type. Without classes, creating a new instance of a certain entity would necessitate rewriting the code for said entity.

Cons

- It is often the case that classes interact with one another. Because of this, if one change occurs in a specific class, other classes that have interacted with said class may need to be modified to reflect the change. This makes adding changes to the program difficult at times, as multiple classes may need to be modified for one simple change.
- For smaller projects, OOD may overcomplicate the code. This is because smaller projects may require only one instance of a certain process and may have that process be very simple. Creating a new class for a very simple process that would only be instantiated once may be unnecessary and may overcomplicate the code.

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 offers minimal comments, besides the comments that are provided by default. The only non-default comments seem to be a block of code that had been commented out in the “DrawScreen” function of the “Project.cpp” file, as well as abrupt comments in the food class and player class. Said comments provide only some information on the functions in which they appear; however, due to the lack of comments in other functions, it can be difficult for someone who has not done this project to understand the code functionality.

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.

The code for the most part has good readability throughout the program, but readability could have been improved in certain functions of certain files. All the class files have good readability, as there is an appropriate use of whitespaces, indentations and newlines. However in the “Project.cpp”, the code within the functions specifically “DrawScreen” there are massive chunks of code without any space between some of the blocks. For example if a specific block is completed like an if statement or for loop, there should at least be a gap of a line or two between the specific block and the next, or if there are nested blocks, the code between the current block and the next block should be provided with space between the two, so that it can be more visible.

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 snake game is not buggy. There are no glitches to be observed and the game does end if the user snake collides with itself. Furthermore, the superfoods implemented do provide the functionality they were intended to provide. For example, the “!” food provides no increase in length or points, the “+” provides an increase in the snake's length, and the “$” provides an increase in points.

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 snake game does not cause memory leaks. We confirmed this by running Dr. Memory. There is no memory leak because the group has ensured that they delete every pointer to which memory is allocated. For example, in the main program, the group allocates memory to the following pointers: game, myPlayer, and myFood. Then, in the CleanUp() function, they proceed to delete every one of the previous pointers. In addition, in the Food class, the group allocates memory to the following pointers: foodBucket and powerFood. They then proceed to delete the previous pointers in the class’ destructor, thus avoiding any memory leaks. Furthermore, in the Player class, memory is allocated to playerPosList, then the pointer is deleted in the class’ destructor. This also happens in the objPosArrayList class, where the group allocates memory to aList and then deletes said pointer in the class’ destructor.

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.

This was an enjoyable experience. We investigated OOD and simultaneously constructed an interesting game. We think the project would have worked better if we were able to construct our own classes rather than have classes provided by default as it would have made the code more personal and unique. This could have also made it easier to program the code using object oriented programming as it would have familiarized us more with the classes, their interactions with each other, and their functions in the main code.