COMPENG 2SH4 Project – Peer Evaluation

Your Team Members	Deyontae Patterson	Victoria Black
Team Members Evaluated	Jack	Anuja

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.

   For the food file, everything is titled well and easily readable and comprehensive. Additionally, code is well organized with comment explaining code that may seem a bit longer or more difficult to comprehend as to why it was necessary. However, as they only have one food object, having the destructor seems a touch unnecessary. The file for GameMechs, there is slightly unneeded spacing between the code in the functions and the ending bracket for the function, Otherwise, things are well organized, and the functions give clear ideas what the function is meant to call and set. These same ideas apply to the file ObjPos.

   For objPosArrayList, They seem to have used an unnecessary else as they could've solely used an if statement with sizeList != 0, as well the way this functions works, only makes it appear as if the tail has been removed, but has not actually been removed. Finally the last file besides the project main one, is the player file, which was really well written, it is organized, and the functions all seem to be written with only relevant statements.

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.

   The beginning is well setup, calling all necessary files and defining all global classes and cases. Then, we move to initialize, they define the size to be 30x15, and calling the classes to create the snake and food ensuring that the player is not colliding with the food as you can see the that you generate the food using the player location as a factor. The run logic ensures that you can see how the player's body interacts with itself, and as well how the player interacts with the food.

   The draw function, is done in such a way that when you read it you can understand that you are going through each column through each row by the nested for functions, then you print each part of the snake one at a time based on location and length, then finally they print all other features once the snake is drawn such as food and the game frame. Additionally, they have a test case for game ends and a screen for it. The final two are to clean up and delay for running organization sakes.

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

   - Class organization
   - Easier setup
   - Classes allow different sections to be changed without altering other functions

   Cons

   - More files required for readability
   - Harder to modify once written
   - Writer needs more self discipline because slightly less enforcement for types

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.

   There is not many comments to thoroughly explain why the code is written the way it is, most of the comprehension comes from the readability and understanding of the person that it reading it. However, in the lack of comments, the code was written in such a way that it is quite comprehensive, as the different classes and functions have clear titles and uses. There are few shortcoming besides slight readability issues such as unnecessary lines, and functions written in roundabout ways such as using if and else with only one case and one of the cases not having any running code.

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.

There is good indentation, there is no case in which we could find an incorrect indentation. Additionally, there was good newlines however a few cases in which there were slightly too many new lines or white spaces. However, the comments were placed well for comprehension where they were placed. Finally, in terms of organization and visual comprehension, they had well organized codes and classes, There is little we would do to improve this besides reducing some newlines.

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)

From playing the game a few times over, occasionally the food will lag slightly and increase the snake by more than once at a time. Additionally, occasionally the key press will not be recognized by the code. Both of these may be due to delay being too large, as you will not rerun the codes until the delay is over which means character press will not be recognized for this time period. However, as this is not a delay that they had chosen, it is merely feedback if given the chance to make this code on their own without guidelines. Otherwise, the game works quite smoothly, the player moving smoothly and always looping around at the correct location. There runs no bugs, and we have tested up to a score of 20.

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.

There is no memory leaks, as shown by the drmemory command in the command prompt when run with their file and game.

```
~~Dr.M~~
~~Dr.M~~ Error #6: UNINITIALIZED READ: reading register eax
~~Dr.M~~ # 0 cmd.exe!?                          +0x0      (0x0048f957 <cmd.exe+0xf957>)
~~Dr.M~~ # 1 cmd.exe!?                          +0x0      (0x00493263 <cmd.exe+0x13263>)
~~Dr.M~~ # 2 cmd.exe!?                          +0x0      (0x0049bcf2 <cmd.exe+0x1bcf2>)
~~Dr.M~~ # 3 KERNEL32.dll!BaseThreadInitThunk +0x18     (0x75d87ba9 <KERNEL32.dll+0x17ba9>)
~~Dr.M~~ Note: @0:00:01.446 in thread 28256
~~Dr.M~~ Note: instruction: cmp    %eax %ecx
~~Dr.M~~
~~Dr.M~~ ERRORS FOUND:
~~Dr.M~~       0 unique,     0 total unaddressable access(es)
~~Dr.M~~       5 unique,     9 total uninitialized access(es)
~~Dr.M~~       1 unique,    60 total invalid heap argument(s)
~~Dr.M~~       0 unique,     0 total GDI usage error(s)
~~Dr.M~~       0 unique,     0 total handle leak(s)
~~Dr.M~~       0 unique,     0 total warning(s)
~~Dr.M~~       0 unique,     0 total,      0 byte(s) of leak(s)
~~Dr.M~~       0 unique,     0 total,      0 byte(s) of possible leak(s)
~~Dr.M~~ ERRORS IGNORED:
~~Dr.M~~       8 potential error(s) (suspected false positives)
~~Dr.M~~          (details: C:\Users\Victo\OneDrive\Documents\DrMemory-Windows-2.5.0\drmemory\logs\DrMemory-cmd.exe.3394
8.000\potential_errors.txt)
~~Dr.M~~       3 potential leak(s) (suspected false positives)
~~Dr.M~~          (details: C:\Users\Victo\OneDrive\Documents\DrMemory-Windows-2.5.0\drmemory\logs\DrMemory-cmd.exe.3394
8.000\potential_errors.txt)
~~Dr.M~~      75 unique,   152 total,  24276 byte(s) of still-reachable allocation(s)
~~Dr.M~~          (re-run with "-show_reachable" for details)
~~Dr.M~~ Details: C:\Users\Victo\OneDrive\Documents\DrMemory-Windows-2.5.0\drmemory\logs\DrMemory-cmd.exe.33948.000\resu
lts.txt
###############################
```

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.

Do to being partners with a friend, we were able to work on the program together in room bookings as to reduce any likelihood of issues with the git push and pull, we shared one computer and were able to diagnose and address errors as a team. This contribution as a team means that we were able to quickly write the pseudocode and locate issues together by working together to figure out what certain variables values should be and locating them with GDB. Occasionally there was concern about being unable to meet at certain times, however during these occasions, we found other times that worked for both partners, or agreed that if one person were to work longer on this one day while the other had another assignment to do, then the other partner would make up for it by doing more work another session.