COMPENG 2SH4 Project – Peer Evaluation

Your Team Members Andy Ngo, Vishva Madu

Team Members Evaluated Everson Shen, Mohamed Ali

Provide your genuine and engineeringly verifiable feedback. Ungrounded claims will lead to deductions.

Part I: OOD Quality

1. **[5 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.
   - Through looking at the varying header files of each object, the given functions can be somewhat easily interpreted what these do within the program due to the simplistic and easily understandable function names; such as getBoardSizeY() where without looking through the given function, you can interpret what this function does due to its given name. An obvious positive feature of these objects within the header files is the easily interpretable naming scheme of the given function. While the simple function names are a nice pro, a negative feature of this is not being able to fully understand what goes within some of these function by just looking at the header. Without understanding fully of what and where each function is used within the program, knowing what specifically goes within the function can be hard to interpret. For instance, the function getObjPos(objPos &returnPos) found within the objPos header file can confusing to know what goes within the function, while you can interpret it will get you some position, you won't fully understand where the function will be used and what specifically will go in the function except for some object. With the lack of comments, it becomes harder to interpret where these functions will be specifically used and what goes within it with just looking at the header files.

2. **[5 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.
   - It's easy to interpret how the objects interact with each other in the program logic because of good naming conventions. The code in structured in an organized manner making it easy to read through the lines.

   - While some may say the code is structured in an organized manner, with the lack of comments, the viewer of the code will either have to interpret how each used functions worked using their own imagination or having to flip back and forth between the different cpp files to understand them. In addition, areas with commented out lines of the program may confuse the reader of the purpose for having them, making it look less professional in the long run if the code is said to be finalized.

3. **[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

- Encapsulation – able to hide data as private so that it's not accidently redefined

- Inheritance – derive subclasses from super classes to handle different cases

- Faster and easier to execute

-Less reusable code, shorter development time

- Easier to assign data and functions to objects under a class

Cons

- Manually controlling memory allocation (pointers)

- Complexity in managing objects and their relationships

- Overhead – slower performance

- Lack of comments to help interpret the code

Part II: Code Quality

1. **[4 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.

- Throughout the code, no sufficient or any comments were to help understand the functionality of the assignment. While there are areas where some of the sections contain commented out code, like within the project.cpp, and words used as pointers to improvements for the implementors, no other areas contain any visual help for other viewers to understand the functionality of the given functions. With this shortcoming of no given self-documenting coding style to help for interpretations, a clear first improvement for the coder is to add comments for the viewer to help them understand what task each function does and where is it specifically used to help run the required code.

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.

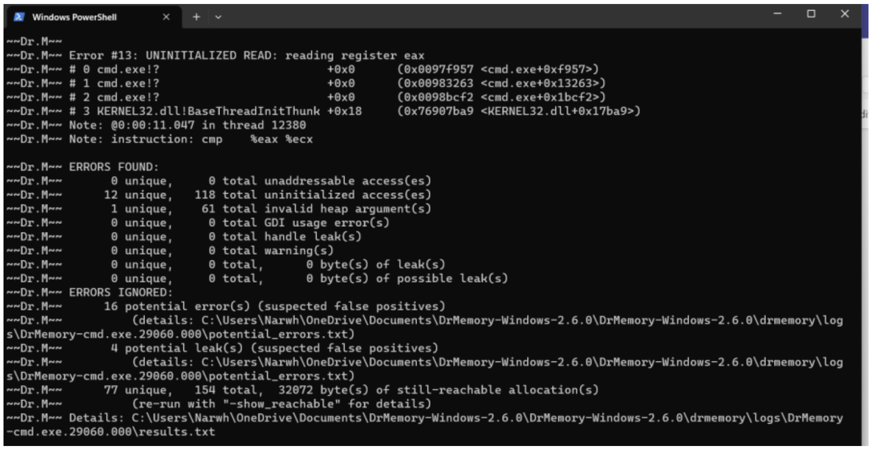
   - While where some areas contain fine indentation and sensible white spaces, areas that contain loops or conditional statements, such as check food consumption function, are a bit clustered as those areas contain a lack of additional newline formatting which makes it a bit harder to read. To declutter the areas where loops are contained, addition new line spaces should be added to improve readability.

Part III: Quick Functional Evaluation

1. **[6 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)

   - With an enjoyable player interface, the snake game offers a smooth bug free playing experience where no noticeable bugs can be found. Through the additional game feature of having more than one hittable object, one extra snake adder and one snake delete block, the game is more fun than the original. While not being able to complete the game fully due to the lack of skill to finish the game completely, the only noticeable feature that seems a bit glitchy is the terminal of the program blinks and flickers. This may be caused due to the large uses of the conditional statements and loops constantly running endlessly until the program is terminated. A solution is to decrease the number of loops and/or conditional statements/

2. **[4 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 areas of leakage were found within the other's team code as their code was free of leakages.



```
~~Dr.M~~
~~Dr.M~~ Error #13: UNINITIALIZED READ: reading register eax
~~Dr.M~~ # 0 cmd.exe!?                        +0x0      (0x0097f957 <cmd.exe+0xf957>)
~~Dr.M~~ # 1 cmd.exe!?                        +0x0      (0x00983263 <cmd.exe+0x13263>)
~~Dr.M~~ # 2 cmd.exe!?                        +0x0      (0x0098bcf2 <cmd.exe+0x1bcf2>)
~~Dr.M~~ # 3 KERNEL32.dll!BaseThreadInitThunk +0x18     (0x76907ba9 <KERNEL32.dll+0x17ba9>)
~~Dr.M~~ Note: @0:00:11.047 in thread 12380
~~Dr.M~~ Note: instruction: cmp    %eax %ecx

~~Dr.M~~ ERRORS FOUND:
~~Dr.M~~       0 unique,     0 total unaddressable access(es)
~~Dr.M~~      12 unique,   118 total uninitialized access(es)
~~Dr.M~~       1 unique,    61 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~~      16 potential error(s) (suspected false positives)
~~Dr.M~~          (details: C:\Users\Narwh\OneDrive\Documents\DrMemory-Windows-2.6.0\DrMemory-Windows-2.6.0\drmemory\log
s\DrMemory-cmd.exe.29060.000\potential_errors.txt)
~~Dr.M~~       4 potential leak(s) (suspected false positives)
~~Dr.M~~          (details: C:\Users\Narwh\OneDrive\Documents\DrMemory-Windows-2.6.0\DrMemory-Windows-2.6.0\drmemory\log
s\DrMemory-cmd.exe.29060.000\potential_errors.txt)
~~Dr.M~~      77 unique,   154 total,  32072 byte(s) of still-reachable allocation(s)
~~Dr.M~~          (re-run with "-show_reachable" for details)
~~Dr.M~~ Details: C:\Users\Narwh\OneDrive\Documents\DrMemory-Windows-2.6.0\DrMemory-Windows-2.6.0\drmemory\logs\DrMemory
-cmd.exe.29060.000\results.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.

   We were able to apply the concepts and knowledge we learned in the course to get our project to work successfully. The project preparation activities also helped made parts of the project simple to implement. It was good working with a partner to discuss ideas and figure out bugs. A challenge we faced during the project was getting the objects and relationships to interact which took some time given that there were multiple files. In conclusion, experience with this software development project was educative and rewarding.