COMPENG 2SH4 Project – Peer Evaluation

Your Team Members Zhihao Zhou and Xuan Chen

Team Members Evaluated Roxxannia_Annie

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

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.
   Pros: From reviewing the head file, all the objects are named appropriately, structure is pretty straight forward, which is easy to understand the behaviour of the function directly from the name. They leave some simple and general comment to understand the behaviour of each element easier.
   Cons: I believe the only part need some improvement is the comment, if there is more explaining comment can help me to read the head file even faster, and this can also help themselves for maintenance the function in future.

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.

```
58      // objPos playPos;
59      // objPos foodPos;
60      // myPlayer->getPlayerPos();
61      //bonus fixing
62      // myFood->generateFood(playPos);
63      myFood->generateFood(myPlayer->getPlayerPos());
64
65  }
66
67  void GetInput(void)
68  {
69
70  }
71
72  void RunLogic(void)
73  {
74      // temp variable to store food position
75      // previous iteration
76      // objPos tempFood;
77      // myFood->getFoodPos(tempFood);
78
```

   In the main program, it is simple and easier to read and understand, the loop is intuitive. The only thing can be improved is the part in the screen shoot, seems they had comment out some code they do not need, I personally believe it is better to delete it since it cause me some convenient when reading it. The other is in cleanup function in the end, I think use else can be better than elseif.

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.
   To summary, the pons and cons are as follow.
   Pros:

A good head file defines the name and function of the object.
The class can be used easily and in multiple places.
Easy understanding name make multiple people work at the same project easier.

Cons:
Slow and use more memory.
The work between different class can be confusing.

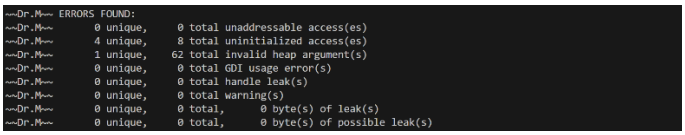
Part II: Code Quality

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 give sufficient comment but I still hope there can be more and explain in detail, in this way the reading of the code for me can be faster, for self documenting coding style, this code has a sufficient delf document coding style but I still hope there will be more comment in the code, this is a efficient way to improve the speed of reading the code. And other part can improve already mentioned before.

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.
   Mainly everything is in the class which make the main function neat, all the white space making the code more readable, after a second read I believe these spaces is as important as the comment in the code, I believe this code did a perfect job on this part.

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 game offer a smooth and bug free experience, all the required function in the project manual has meet in the code, the snake can show up from the other side of the gameboard after it hits one side, all the food is consumed in a right way, normal food give score and length, bonus give the score only, and self collision can be detected and successfully terminate the game, also space bar can end the game too. All in all, this is no problem I can see here.
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.



From the report seem there is no leakage at all, they did a great job. But I do have one thing I don't understand, there is not delete for new, why there is no memory leakage?

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.
   Working on a new language is not easy, it takes time to familiar with it, this process is slow. Since I am not familiar with the language, little mistakes are one after another and these small problems are not easy detected and cause a lot of the lose of time. Work in a team need communication, otherwise there is a lot of misunderstanding and both side need time to familiar with the other's work. All in all, this is a good experience.