COMPENG 2SH4 Project – Peer Evaluation

Your Team Members ____Xiaojun Shen____ ____Tingyu Meng____

Team Members Evaluated _____Tina_____ _____ Chloe _____

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 files are clean and simple and are keeping an intuitive naming convention for both the class and class functions, which makes it easy to interpret the function inputs and outcomes, one minor problem is that it's lacking comments to explain some specific function behaviors.

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.

   It is quite easy to interpret the relationship between the classes, as the name of the class pointers are informative, and the code is generally well organized. And there are also comments that explain the temporary storage objects and content comparison.

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
   - **Makes the functions and variables more organized.**
   - **Increases the readability.**
   - **Increases the maintainability, easier to debug.**
   - **Easy to add new structures.**
   - **Avoid repetitive coding.**

   Cons
   - **Makes the starting stage of the coding process less intuitive.**
   - **Sometimes can make the coding process 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.

   Yes, the code offers sufficient comments on almost all the self-written code, great and brief explanations are provided, I can understand the code functionality more efficiently with the help of the comments. The only shortcoming that I can think of is the comments only provide a brief explanation on the functionality, but the methods used to achieve the purpose are not described.

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.

   Yes, most of the code follows good indentation and provides great readability, except for some parts in player class file and project.cpp file. In the special food consumption part in player class, the multiple scores increment process include too many repeated codes, I would use a for loop to decrease the code size and provide better readability, the eat food logic part is also a little crowded.
   On the other hand, the variable declaration part of the Drawscreen() function in project.cpp file is crowded, though well commented, still have some trouble reading.

```
// Variables for iterating through grid
int i, j;
int x = snakeGameMech->getBoardSizeX();
int y = snakeGameMech->getBoardSizeY();
objPos tracker;
// Variables for iterating through "snake list"
int s;
int snakeList = (snakePlayer->getPlayerPos())->getSize();
objPosArrayList* snakePtr = snakePlayer->getPlayerPos();
objPos snakePos;
// Variable for holding food position
int f;
objPosArrayList* foodPtr = snakeFood->getFoodPos();
objPos foodPos;
// SUPER Food! -> score +5, increase snake size +1
case '$':
    mainGameMechsRef->incrementScore();
    mainGameMechsRef->incrementScore();
    mainGameMechsRef->incrementScore();
    mainGameMechsRef->incrementScore();
    mainGameMechsRef->incrementScore();
    playerPosList->insertHead(newHead);
    break;
```

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)

   Yes, the snake game offers smooth, bug-free playing experience. But on one of our teammate's MacBook the program always ended with a error saying "pointer being freed was not allocated", we discussed and thought maybe there was a double freeing situation happening in the code. The possible root cause is maybe some variable was passed into function by value instead of by reference, the value would first be deleted at the end of the function, then the destructor would delete it again, which could cause double free error. Though the program is still smooth, but it would be better to fix the error.

```
project(29678,0x1db9d9ec0) malloc: *** error for object 0x137606060: pointer being freed was not allocated
project(29678,0x1db9d9ec0) malloc: *** set a breakpoint in malloc_error_break to debug
zsh: abort      ./
project
```

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 doesn't cause any memory leak. No memory leakage takes place is because the code follows a great habit of adding delete() to the corresponding "new" variables. The code allocated heap memory in Food() class constructor, objPosArrayList() class constructor and Player() class constructor respectively, and delete() function is called in corresponding destructors. The code also allocated heap memory in Initialize() function in project.cpp, but the heap memory is also deallocated by using delete() function in cleanup() function of project.cpp.

   But there is still a heap memory related problem taking place in the program as illustrated in the former question. Some heap variable is freed twice in the code, which is not a good behaviour of programming.

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 first collaborated software development experience is great, I felt like learning a lot from the other's programming idea. On the other hand, this first time working with others on the same program is also a challenge, because every person has different coding habits, we must understand each other's thoughts and then develop the code. All in all, programming in groups actually makes me spend more time on thinking through the code functionality, but it also provides me with more opportunities to learn from other students.