COMPENG 2SH4 Project – Peer Evaluation

Your Team Members Ziyi Chen , Zijie Zhai

Team Members Evaluated Josh Crosby , Andrew DePetris

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.

   The OOD header files of their group are very well designed. it is easy to identify the possible behavior of objects and their interactions in the project. Each object's responsibilities are clearly defined, and it has explicit attributes that indicate its role and function. The separate design of the Food Class also makes debugging the program much easier and more understandable.
   There isn't and shouldn't be a negative side on OOD design in this "small" program.

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.

   Due to the appropriate use of pointers and Temp values. It is not difficult to read the code after checking the h-file. You can easily understand the interaction and division of labor through pointers and functions. However, although it is not difficult to read, more frequent comments to the code will make the reading time shorter.

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.

   As opposed to the procedural programming approach in C in PPA3, the use of OOD in Project allows the code to be split into multiple objects for division of labor. This makes it more easier to read the code, and it helps multiple engineers to work together on the same project, so that they can work on their own objects before integrating the final program.But it will increase the Project size and take more memory during running.

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 has almost no coment, but due to the simplicity of the OOD design approach and the size of the functions, reading most of the C++ documentation is not very difficult. However, for Project.cpp and objPosList.cpp, the lack of code comments makes reading slower. I think basic commenting is still necessary for large files , it can greatly increase reading speed.

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 indentation are generally good, but in some parts it not that perfect and might cause some tiny reading problem.

```
bool playerElement;
objPos temp;
playerBody->getHeadElement(temp);
objPos foody;
myFood->getFoodPos(foody);
int size = playerBody->getSize();
for(int i = 1; i< size;i++)
{
    objPos colPos;
    playerBody->getElement(colPos,i);
    if(temp.x == colPos.x && temp.y == colPos.y)
    {
        myGM->setExitTrue();
    }
    }
if( temp.x == foody.x && temp.y == foody.y)
{
    playerBody->getHeadElement(temp);
    playerBody->insertTail(temp);
    playerBody = myPlayer->getPlayerPos();
    myFood->generateFood(playerBody,x,y);
```

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 code runs very smoothly and without any bugs.

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.

   Base on the Dr Memory Report, There is no memory leak because of all the memory used on heap are freed in the destructer under objPosArrayList.cpp and player.cpp.

```
ERRORS FOUND:
     0 unique,    0 total unaddressable access(es)
     8 unique,  116 total uninitialized access(es)
     0 unique,    0 total invalid heap argument(s)
     0 unique,    0 total GDI usage error(s)
     0 unique,    0 total handle leak(s)
     0 unique,    0 total warning(s)
     0 unique,    0 total,     0 byte(s) of leak(s)
     0 unique,    0 total,     0 byte(s) of possible leak(s)
```

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.

   The experience on our first collaborated software development project is pretty good. Collaboration between multiple engineers on OOD project design can greatly reduce the time engineers need to spend reading/understanding the code. And it can also be debugged more conveniently. Completing code with team members is easier than the previous PPA and Lab, because when encountering bugs or design difficulties, you can communicate with each other to get inspiration。