COMPENG 2SH4 Project – Peer Evaluation [25 Marks]

Your Team Members: Team: DJ

Members (Name, MacID, github user):

Dhushan Kiritharan kirithd (kirithd-2024)

Jacob Dias diasj11 (diasj11)

Team Members Evaluated: Team: teamsize != 2

Membners : Narends and melnbaa

Provide your genuine and engineeringly verifiable feedback. Ungrounded claims will lead to deductions. Completing the peer code evaluation on time will earn your team a total of **25 marks**. Do not exceed 2 paragraphs per question.

Peer Code Review: OOD Quality

1. **[3 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 program shows a very clear and organized object interaction flow. It utilizes the functions within GameMechs and the Player classes effectively to access different elements, clearly organizing functions to return variables that are needed with accurately named functions. The logical flow is quite strong and is easily interpreted. However, there are no use of comments in the main file, which would enhance the readability of the code significantly for users who are unaware of the functions of the game.

2. **[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.

   In terms of pros, the code that utilizes the OOD approach is very easy to implement new features and build upon with additional functionalities, as by having defined classes that tackle different functionalities separate from other classes. However, it does increase the number of files and the complexity of the code in terms of stringing all the classes together for an object to be able to undergo the essential functionalities.

Peer Code Review: Code Quality

1. **[3 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.

Certain files lack any comments such as the Project.cpp file and other files have very limited comments. Although the code decisions that are not initially obvious are often commented, it would be in the codes benefit to provide comments that outline the general functionality of functions, this making it so the code can quickly be processed.

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.

   The code utilizes very clear indentation and sensible white spaces, making it very clear how if statements and other structures are formatted, as well as using new line commands to properly format the various outputs effectively.

Peer Code Review: Quick Functional Evaluation

1. **[3 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 technical user feedback)
   Upon a death, instead of presenting a loss message, the following error message happens instead:

   Project(23955,0x20263b240) malloc: Double free of object 0x12e704080
   Project(23955,0x20263b240) malloc: * set a breakpoint in malloc_error_break to debug**
   zsh: abort ./Project

   This error also sporadically occurs in regular gameplay upon a random movement.

   This error seems to be stating that there is a pointer being freed more than once or freeing memory from a pointer that isn't dynamically allocated. To solve this issue, note where memory is being allocated and where delete or free is being used and use stepover/stepinto/stepout to observe when and if a pointer is being freed correctly.

2. **[3 marks]** Does the Snake Game cause memory leak? If yes, provide a digest of the memory profiling report and identify the possible root cause(s) of the memory leakage.

   There are no memory leaks.

Project Reflection

Recall the unusual objPos class design with the additional Pos struct. After reviewing the other team's implementation in addition to your own, reflect on the following questions:

1. **[3 marks]** Do you think the compound object design of objPos class is sensible? Why or why not?
   I don't think its sensible because it is not necessary to implement dynamic memory management for the purposes of the games x and y position coordinates. This is simpler and faster as well.
2. **[4 marks]** If yes, discuss about an alternative objPos class design that you believe is relatively counterintuitive than the one in this project. If not, explain how you'd improve the object design.

You are expected to facilitate the discussion with UML diagram(s).

Instead of dynamically allocating the variables, it can simply be stored as member variables within the objPos object and have functions that return these values, so they can be accessed not during the entire runtime, but only when it has to be accessed. The structure as indicated in the following UML diagram is much more efficient