|  |
| --- |
| Code Review Report |
| Project Summary |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| Report Date | Project Name | Student Name / Number | Code Reviewer |
| *Date* | *Project* | *Name* | *Luke Moseley – s4203133* |

|  |
| --- |
| Project Summary |
|  |
| ***Description here…*** |

|  |
| --- |
| Unit Testing |
|  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Summary | Test Steps | Expected Result | Actual Result | notes |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

|  |
| --- |
| BUG Tracking LOG |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| Bug Details | STEPS TO Reproduce | DATE Found | Date Resolved |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |
| --- |
| Conclusions/Areas for further expansion |
|  |
| ***Conclusions here…*** |

|  |
| --- |
| Project Checklist – (For Code Reviewer to fill in) |
|  |

|  |  |
| --- | --- |
| task | notes |
| *What is the code supposed to do, and does it accomplish it? (Is the code easy to understand?)* | **The code provides flocking behaviours to a group of boids, as well as the option to evade from a nearby persuer. The code allows for boids behaviour to be defined, with a switch case efficiently handling that. There are bound checks to keep the boids within a confined space, and the ability to flee from nearby obstacles. At the moment, the boids sometimes seem to fly through the obstacle, but this may be something that is fixed at a later time.** |
| *Can this solution be simplified? (How?)* | **Some areas of code could maybe be split down into differnet functions, such as the behaviour switch case, where the functionality of each case could be put into it’s own function, allowing for more modular and reusable code.** |
| *Can this solution be improved in terms of maintainability, readability, or performance?* | **Same as above. Another suggestion would be to change how the boids are aboiding obtacles. As the functionality for avoiding an obstacle is very similar to the seperation behaviour, polymorphism could be used to create different types of the same seperation function, but pass in different variables. For example, the default seperation function could cause boids to flee from other boids, another version could have obstacles passed in instead so boids avoid obstacles, and so on with different types of objects.** |
| *Is the code modular enough? How reusable is it?* | **The different boid behaviours are split into their own functions allowing them to be re-used in other places.** |
| *Are there any best practices or design patterns that could improve this code?* | **Some optimisations techniques could be included, allowing for more boids to be involved in the simulation. Maybe an object pool could be used for spawning obstacles, although this isn’t necessary as there is only a small amount is being spawned.** |
| *Does this code adhere to Object-Oriented Principles, like the Single Responsibility Principle?* | **From what I’ve seen, yes. I can also see that C++ functions have been made into blueprint nodes, allowing C++ functionality to be accessed from blueprints, which is a nice feature.** |
| *Can you think of any use case in which the code does not behave as intended?* | **The boids don’t always avoid obstalce, and sometimes fly straight through. Maybe the avoid function needs to be looked at?** |
| *Are debug-logs user friendly?* | **Yes, the debugs that I have seen have made it clearer to see what’ss going on.** |
| *Does the code contain user friendly comments or documentation? (Is it up to date?)* | **The comments that are included describe what is happening well. Not all areas of the code has been commented, but this may be due to the project still being work in progress, so I can imagine more comments will be added later on to help document all of the included features.** |
| *Does the code make use of user data in a way that might raise privacy concerns?* | **Not that I can see.** |
| *Is the code ethical? Does it exploit behavioral patterns, introduce bias, or have the potential to cause harm?* | **Yes the code is ethical, I cannot see how it may do harm.** |
| *Is the code testable? If so, what automated tests have been added?* | **Different aspects of the boids behaviour can be customised through the UI, so the boids can be tested out using different values. This allows for testing out how the boids behave under different circumstances.** |
| *Are there portions of code that aren’t covered in testing?* | **Blueprints weren’t covered, although blueprints weren’t the main focus of the assignment.** |
| *Were any parts of the code confusing or difficult to understand?* | **Not from what I saw.** |
| *Any final comments or feedback:* | **This seems like a good boid simulation. With some extra features and polishing, and the fixing of some small bugs, this project can developed further into a great simulation that demonstrate more advanced functionality and interesting boid behaviour.** |
| *Signature/Date:* | **Luke Moseley – 12/12/23** |