Augustus Mendy

Southern New Hampshire University

CS-499-10450-M01

Professor Ramsey

September 14, 2025

2-1 Journal: What Makes a Productive Code Review

**Part 1:**

1. **What is code review?**

Code review is a methodical analysis of computer source code that aims to find any problems, improve code quality, and make sure that best practices and coding standards are followed. Peers or senior developers examine the code thoroughly during this step and offer helpful criticism on its correctness, efficiency, security, and clarity.

1. **Why is it an important practice for computer science professionals?**

Given that code reviews greatly improve software's overall quality, maintainability, and stability, they are crucial for computer science professionals. Furthermore, by identifying possible mistakes early in the development cycle, code reviews assist prevent the introduction of security risks, encourage consistency across projects, and facilitate knowledge sharing among team members. Peer review enables you to find and fix mistakes in your code immediately. Allowing a peer to review your code is a fantastic method to learn about their viewpoint. Reviewing the code with multiple eyes is preferable to doing it with just one.

1. **What are some code review best practices that you read about in the resources that are crucial to include in a code review? Include when a code review should occur in the development process with a rationale as to why.**

Making sure unit tests are included in the code and operating correctly is, in my opinion, a great practice. I am aware of my propensity to duplicate code. Therefore, another recommended practice is to make sure that there is no repeating code. And if I had to choose just one more, I would answer that I would make sure the code adheres to coding standards. Furthermore, who could overlook security? Does your code have security? I believe that to be the most crucial best practice to promote.

when you are writing code with your team. The code is written by the developer, who also prepares a checklist for the reviewer to examine. Having a code review as soon as the code is written is crucial because coding is so detailed. One excellent resource for developers is GitHub. The code can be uploaded by the developer, and the team can evaluate it, identify any flaws, and immediately fix them. After the modified code has been fixed, they can commit the modifications, making the new code accessible and operational. You want to test frequently, just like in all of my classes. The same is true for code reviews. To ensure that the released code is error-free and operational, review frequently.

**Part 2:**

1. **What software have you chosen to use to record your code review?**

I have chosen to use Visual studio

1. **Describe your approach to creating an outline or writing a script for your code review for each of the three categories that you will be reviewing based on the rubric as well as the code review checklist.**

Introduction:

Three criteria will be used to assess the project in this review: code quality, functionality, and design. Giving helpful criticism in line with the code review checklist and rubric is my aim.

Design:

Good modularity is demonstrated by the program's overall structure, which is divided into distinct files for models, cameras, and shaders. I looked to see if the code made it simple to extend—for instance, by adding additional objects or textures—and it does. Additional abstraction, such as classifying repetitive transformation logic into helper functions, might improve certain aspects of the design.

Functionality:

I experimented with the camera controls in the application, such as zooming, panning, and circling. All of the controls worked without a hitch, although for more accuracy, the zoom speed may be changed. Textures were mapped accurately, and lighting was applied to provide realistic shading effects. Additionally, I tested edge scenarios, such shifting the camera to extreme places, and the software handled them with elegance.

Code Quality:

Since the code adheres to standard naming rules, it is readable. Although most important sections include comments, certain intricate shader logic might need more precise explanations. Additionally, I observed that the render loop contained redundant matrix multiplications that might be streamlined into a single calculation to increase efficiency. Overall, the code is clear and well-organized.

Conclusion:

In conclusion, the project shows good design and functioning, while there is potential for small readability and efficiency gains. The code would become much more extensible and maintainable with these improvements for next developments.