CMSI 371-01
COMPUTER GRAPHICS
Spring 2015

Assignment 0326b Feedback

Outcome *3a* now covers enough of the overall graphics library to merit a full proficiency range. With instance transforms, outcome *3d* now covers the full envisioned vertex shader, and also drops the proficiency cap even with the fragment shader remaining.

Juan Carrillo

juansc / juanscarrillo23@gmail.com

Note that a test suite for your matrix library was a major component of this assignment. No such thing could be spotted among the submission's files.

1. This function should be gone from the main code, instead implemented now by the matrix library. (*4b*)
2. OK, so some matrix-related tests reside here. However they conflate `Shape` object functionality with the matrix functionality, so they do not really serve as ideal *unit* tests for just the matrix library. There should be a separate suite that tests *only* the matrix code. (*4b*)
3. Even though this file defines a top-level scope object, it is still good practice to enclose the definition code inside a function, for proper encapsulation. (*4b*)
4. Ah see, you *did* refactor this function into your matrix library. Out with the old one! (*4b*)

2a — **+**

2b — **+**

3a — **|** …A robust unit test suite is a key part of a solid matrix library.

3d — **+**

4a — **|** …The missing test suite hits here also, because the stability of the matrix library's functionality is at greater risk.

4b — **|** …This is primarily the single missed rotation-by-arbitrary-axis refactor indicated in notes #1 and #4. Although the proficiency drop may seem disproportionate to the indicated issue, I would argue that it *is* a big deal—when refactoring code, you should be very clear about what should get moved and why. If anything, let this be a reminder for greater awareness of this in the future.

4c — **+**

4d — **+**

4e — **+**

4f — **+**