CMSI 371-01

COMPUTER GRAPHICS

Spring 2013

Assignment 0404 Feedback

The "cuffs" are off outcomes *2a*, *2b*, and *3d* with this assignment, so I have started giving **+**'s for those outcomes if the submitted work calls for it. *1c* and *3a* await full scene interaction before they can max out. *2c* (not part of this assignment) and *3e* need proficiency in lighting and fragment shaders in order to go **+**.

Quin Thames

1c — Your instance transform functionality has given you a new level of flexibility in composing and arranging objects in your scene. (**+**)

2a — Instance transforms === full transform application proficiency. (**+**)

2b — Projection has been successfully implemented. (**+**)

3a — Nice transition of rotation from the global matrix to the use of your instance transform. Keep this up and this outcome will turn out fine. (**|**)

3d — Your library is pretty much complete, with the exception of the camera matrix. And you should revisit your unit tests—they need some attention from your last updates! (the knock in the proficiency is mainly because of that, and not the missing camera matrix) (**/**)

3e — You have successfully extended your vertex shader to use instance transforms. Keep it going and you will cruise nicely to a **+** when all is said and done. (**|**)

4a — The biggest knock on your code's correctness is the regression in your unit tests, and this is mainly because the test suite appears to not be kept in sync with your library. You should get into the habit of running your test suite as your first stop whenever you change a library. (**|**)

4b — Your code looks properly structured and separated; the pulling out of `objectsToDraw` is noted and justifiable based on its size and complexity. (**+**)

4c — Your code shows decent readability but as usual there are places without spaces where I think there should be (that sounds Dr. Seuss-ian—maybe this will make it easier to remember? **>:-)**). (**+**)

4d — Your work certainly shows good information and resource use. (**+**)

4e — Your commit frequency and messages provide an excellent record of how your code has evolved. (**+**)

4f — Submitted on time. (**+**)