

CMSI 371-01

COMPUTER GRAPHICS

Spring 2013

Assignment 0326 Feedback

For this assignment, outcomes *2a*, *2b*, *3d*, and *3e* max out at | because the requested functionality in this assignment do not yet reach the culmination of what these outcomes represent overall.

Chase Blokker

2a — You’ve taken a few more concrete steps toward full 3D transform proficiency—now on to using these functions in your scene! (|)

2b — The mechanics of your ortho and frustum functions look good, although as you noted, you changed the signs of the κ -associated parameters. Not necessarily wrong and potentially workable, but you will see a difference in projection behavior. The next test will be to use them in your scene. (|)

3d — Your matrix library is certainly moving in the right direction. Actual “field testing” in your 3D scene code is up next. (|)

3e — Your matrices represent additional progress toward 3D scene rendering, but as mentioned will not top out this outcome yet because we haven’t covered the full range of shader functionality yet. (|)

4a — The code that you have works well so far, and this is bolstered by having a unit test suite available to “keep it honest.” (+)

4b — Separation of concerns looks well taken care of in the code that you have so far. My only comment is that some functions are better done in object-oriented style (i.e., they reference the `this` variable) and thus have to be assigned to `Matrix4x4`’s prototype. Matrix multiplication is a prime example of this, and in your code, `toWebGLMatrix` is also a good candidate for this kind of approach (see my inline comments). (+)

4c — Your matrix code is quite readable, although it is true that I wrote some of it :) (+)

4d — Your work shows fine resource use, including leveraging the rotation matrix code that is already in the sample programs and the projection matrices that are already in the handouts. Your comments indicating what you refactored some sample code are helpful in tracing your code’s progeny. (+)

4e — Your commit phasing and messages show a decent pace, particularly with the separation of some work into distinct functions. Make sure to establish this as a work habit: write the test; write the implementation; commit when the test succeeds. That gives your commit log a very logical, trackable evolutionary trajectory. (+)

4f — Submitted on time. (+)