

# CMSI 371-01

## COMPUTER GRAPHICS

### Spring 2013

#### Assignment 0404 Feedback

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

#### Kaitlyn Higa

(updated feedback based on your final September 17 commit)

*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* — Orthographic matrix is successfully implemented in the library and used. The frustum matrix is implemented and looks right, but is difficult to test within the context of the scene due to the lack of a working camera matrix. (|)

*3a* — Tweened animation is successfully implemented—you got a good deal of help, but at least you still put it together successfully. The way you use this code in the interactive aspects of your scene will be included in the next assignment evaluation. (|)

*3d* — You’ve put together a decent library of shapes and matrix transformations. The shapes library is nice and robust; the matrix library has the basics except for an unfinished camera matrix. (|)

*3e* — You have successfully extended your vertex shader to use instance transforms. When we factor in the successful fragment shader implementation in the later assignment, you will cruise nicely to a + when all is said and done. (|)

*Because the final three assignments were combined in a single body of work, the proficiencies below are cumulative and apply to your final code base. Later feedback sheets will reiterate the proficiencies given here.*

*4a* — You have packed a good deal of functionality in your scene, but overall the quality suffers from numerous glitches in the code (see inline comments). The fact that some of these glitches render the code unrunnable in its committed form really detracts from the proficiency here. A whole chunk of work and functionality gets seriously spoiled when little bugs keep the overall program from running. (/)

*4b* — Your code looks clean and properly structured. (+)

*4c* — Your code is generally readable and easy to understand. Some of the glitches mentioned in *4a* are purely syntactic, but “forgivable” syntactic (i.e., the code runs anyway). Although this code still runs OK, having too many of such syntactic hiccups may cause confusion and distraction to a third-party reader of your code. (|)

*4d* — You have certainly shown good information and resource use throughout your summer work. (+)

*4e* — Your commit frequency and messages start out in June/July with a very good record of the changes made to your 3D scene. The final commit on September 17, though, shows extensive changes that really should have been done in smaller pieces. (|)

*4f* — Not submitted on time. (−)