

# 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 +.

#### Haley Young

*1c* — Your instance transform functionality has given you a new level of flexibility in composing and arranging objects in your scene. Although, as we discussed, based on the specific needs of your scene, you may want to model your instance transform as an array instead of a single object. (+)

*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, unit tests and all! (+)

*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* — Your code is certainly functional and correct. The biggest knock on it would be a number of design questions that I brought up in the inline comments. However the scene does look like it works as intended, so I will leave this outcome alone. (+)

*4b* — Your code looks mostly properly structured and separated; the one thing that sticks out is your logic for putting together your rotation transforms. That’s some distinctly repeated code, and whether or not you change your instance transform approach to use arrays of smaller transforms, this portion should be factored out into a function anyway so that it can be unit-tested. (|)

*4c* — Your code is quite readable and maintainable. (+)

*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* — Something submitted on time, with some refinements and bug fixes coming in later. (|)