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 **+**.

Britain Southwick

1c — Your instance transform functionality has given you a new level of flexibility in composing and arranging objects in your scene. Although yes, as you noted in your comments, you need to design a better approach for *updating* those transforms over time.

Your implementation of object composition is also noted here, although the implementation is limited to one level of children. Clean that up to max this out. (**|**)

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

2b — Projection has been successfully implemented, but not yet applied. You really were that close—it really could have been done pretty quickly! (**–**)

3a — Some progress here in terms of actual animation, but as you said yourself, you need a little generalization. Expand your model a bit; plus, you may want to separate out the code whose sole job is to *change* the properties of your scene's objects. The interaction and dynamic behavior handout might help. (**|**)

3d — Your library is pretty much complete, including, with my help, the camera matrix. A few more unit test cases—including for the camera matrix!—will help solidify the robustness of your code. (**|**)

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 — With the exception of your one-level composite object functionality, everything looks correct and functional. Your animation hardcoding doesn't drag this one down because that is more of a focus for the next assignment, so this hit is just for composite objects. (**|**)

4b — Your code looks properly structured and separated, except for your composite object implementation which repeats similar blocks of code in at least three different places. (**|**)

4c — Spacing is generally OK except for your transform objects (which will be fine without such deep indents) and the occasional missed indentation (not including mine with the test camera). Readability hiccups seem to correspond to places where you added new functionality—sort of understandable that this is where they are, but still avoidable. Train yourself so that this type of formatting is second nature and something that you can do alongside actual programming. (**|**)

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. (**+**)