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 +.

Carlos Agudo

Because Assignments 0319, 0326, and 0404 all cover the same code base and may have cumulative results, this feedback sheet covers *all* re-reviewed outcomes from these assignments. The only outcome that is unaffected, because it is time-sensitive, is *4f*.

1b — Your shapes library shows a good understanding of how polygon meshes are formed. (+)

1c — You have *an* implementation for composite objects and instance transforms on them, but this implementation is incomplete. Your composite objects cannot go arbitrarily deep, and they way you handle transforms from parent to child is buggy. (/)

2a — As mentioned in *1c*, you have some instance transform capability, but it is buggy. The bugs also include inconsistencies in your matrix library with regard to how you handle transforms. (/)

2b — You have successfully implemented and used a 3D frustum projection on your scene. (+)

3a — Despite bugs in your instance transformation implementation, you have enough to accomplish a degree of 3D animation (which I helped you implement with a keyframe tweener). This animation does not work perfectly well, but it is better than nothing. (|)

3d — In terms of coverage, your matrix library is nearly complete, missing only a camera matrix (which you don’t seem to need anyway). However, it has fundamental bugs, primarily with regard to its [lack of] consistency when handling translation, rotation, and scaling transformations. In addition, you now have *two* copies of your matrix library, and the one that has a unit test is the one that is *not* being used. Sloppy. (/)

3e — Your vertex shader accommodates your current instance transform implementation, although it cannot handle “inherited” or “parent” transforms well. (|)

4a — Your code is peppered with logical flaws and bugs: composite objects, matrix library functions, and instance transforms all have notable issues. The individual problems build upon each other, such that bugs in one make it harder to fix bugs in the other. My suggested “fixing” order is to clean up your matrix library first, then composite objects, and finally instance transforms. (/)

4b — Your separation of concerns is mostly respectable. Your primary technical issues have to do with specific functional bugs and not the overall structure of your code. (+)

4c — Your code has overall good readability, with the main lingering issue being a lack of consistency with spacing, line breaks, and indentation (though not too much as to drag down the proficiency). (+)

4d — You have done a decent job of using available resources and information to provide breadth coverage for the functionalities sought by the assignments. You now need to use resources for better depth so that bugs are fixed. (|)

4e — Your version control habits show good consistency and messages for the overall body of work done for the covered assignments. (+)