CMSI 371-01

Computer Graphics

Spring 2016

Assignment 0329b Feedback

All caps are released with the outcomes in this assignment because a sufficient amount of functionality will have been reached here.

Ed Seim

SirSeim / sirseim@gmail.com

*Notes while running (high-priority notes are marked with ***):*

- Well you get some kind of test suite up…but just one case for most categories?
- The rectangular viewport hints that you've gotten the projection part down…but it will require a look at your code to really see how you've handled *The Matrix*.

Code review (refer to http://lmucs.github.io/hacking-guidelines/ for code-review abbreviations):

1. I see how you are recursively propagating matrix transformations in `Shape`'s `translate`, `scale`, and `rotate` downward to its children, but ultimately that incurs unnecessary computations. As you build a scene, this means that every step that you transform something triggers multiplications over and over again as they get propagated. If, instead, you don't propagate right away but just wait until drawing time to include a parent's transformation, you can potentially save on all of these multiplications. Just a thought. More of an optimization than anything else; I'm satisfied that the understanding is there. (*4b*)
2. Oh and another thing…this approach assumes that the complete object tree is finished before transformations are done. What if transforms are done, and *then* a child is added somewhere down the tree? I don't see any code that ensures application of parent transformations to that child. (*4a*, *4b*)
3. And after all that, I'm not seeing that this propagation is being fully exercised by the scene that you have now. I think it will work, but it's another thing to work with it as you code something up. Hope you are able to get there eventually. (*2a*)

2a — | …Transformation framework looks good in theory (as long as the object tree is completely built first before transformations are applied), but needs practice.

2b — + …How about some frustum action next? **:)**

3a — +

3d — +

4a — | …Potential bug when adding children after a transformation has been applied.

4b — | …Interesting child-transform design, but likely will not scale.

4c — +

4d — +

4e — +

4f — +