CMSI 371-01
COMPUTER GRAPHICS
Spring 2015

Assignment 0326b Feedback

Outcome *3a* now covers enough of the overall graphics library to merit a full proficiency range. With instance transforms, outcome *3d* now covers the full envisioned vertex shader, and also drops the proficiency cap even with the fragment shader remaining.

Joaquin Loustau *joaquinloustau / joaquinloustau@gmail.com*

1. OK, so you have a makeshift viewing volume and "camera" matrix here. Useful at this early stage, but you'll probably want the real thing eventually. (*2b*, *3a*, *3d*, *4a*)
2. Note, this is a candidate for refactoring as a method (a short one, yes, but a method either way). (*4b*)
3. *** Remember that children should inherit whatever transform is currently affecting. (*1c*, *2a*, *4a*)
4. This is pure library—the extraneous *matrix3D-webgl.html* and *glsl-utilities.js* files should be removed from this folder. (*4b*)
5. *** Your projection matrices look right—why aren't you using them? (*2b*, *3d*, *4a*)
6. Good test coverage here; your `Matrix3D` object looks pretty solid—you should use it more! (*2a*, *2b*, *4a*)

2a — +

2b — /

3a — +

3d — +

4a — | …This is mainly the missed detail regarding how the instance transformation should interact with a `Shape`'s children, if any. Also, to a lesser degree, there is the mysteriously unused projection matrices.

4b — + …Yes, there is a candidate `prototype` function, but missing it in this first round isn't horrible.

4c — +

4d — +

4e — +

4f — + …Consideration given for deadline due to spring break travel.

Updated feedback based on commits up to 2015-05-10:

Major points are addressed; your matrix library is now fully leveraged in your scene, especially the instance transformations. You should have gone with a frustum projection though.

2b — + …This is solely for using *any* projection. As you know the issue of not having chosen the *better* projection is noted later.

4a — +