CMSI 371-01
COMPUTER GRAPHICS
Spring 2016

Assignment 0329a Feedback

This is the last assignment where *3a* tops out at **|** as we head toward the full expected functionality of your library code. Meanwhile, *3d* also stays at **|** until we get sufficient functionality in our shaders.

Christopher Dellomes   *cjdellomes / cjdellomes@gmail.com*

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

- Your seen contains a single rotating icosahedron, which doesn't really demonstrate the degree to which you fulfilled (or not) the instructions of the assignment. We'll have to get into the code then.

- The shape unit test suite is noted however, and that is good to see. It runs without failures, though at 10 assertions only it can probably use some improvement in coverage.

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

1. OK, the shape library is present and working. Cylinder + cube + sphere is a decent collection, definitely composable into an interesting scene eventually. (*+1b, +3a*)

2. Child/group object support is also seen, so that's good. I uncommented the `compoundShape` code to see this. I wish this was actually the default scene—they come together nicely! (*+1c, +3a, +3d*)

3. Design- and separation-of-concerns-wise, the vertex-buffering and drawing routines can become `Shape` prototype functions instead, since they will pretty much be the same regardless of what the specific scene might be. (*4b*)

1b — **+**

1c — **+**

3a (max **|**) — **|**

3d (max **|**) — **|**

4a — **+**

4b — **|** …`Shape` object can use a second pass to consolidate additional functionality. Put yourself in the position of having to create a new, separate scene with the libraries you've written. What code gets copied over? Those are likely good candidates for additional functions to fold into your libraries. Ultimately, only the code specific to a particular scene (the "app"), like its actual objects, or how they are updated during animation and interaction, should ideally be in the "app-specific" code.

4c — **+**

4d — **+**

4e — **+** …Decent frequency and descriptive messages.

4f — **+** …Sufficient work done by the due date.