CMSI 371-01

COMPUTER GRAPHICS

Spring 2013

Assignment 0418 Feedback

Outcome *3a* can now "graduate" with this assignment, with *2c* and *3e* maxing out in Assignment 0502.

Chris Whiting

Your code, as of May 4, is currently in flux, so this evaluation is based on an older commit that focuses on the interaction aspect of your scene (specifically, commit `ed75958a43ffdedbd0c2eb2290490fcf10ceefeb`). Thus, there might be some suggestions/feedback here that you have already addressed. If you have, then great; if not, then that is something to consider doing as you wrap everything up for your portfolio.

1c — Your scene/object composition functionality exists in code, but is not being exercised by your scene. You really should do this—stick a few composite objects in there; for example, you can make your zombie a composite rather than just a single sphere (e.g., give it arms/legs, a head, etc.). This will show you whether or not your recursive calls are working. It remains a tad repetitive (i.e., you have blocks of code that appear twice when, with proper structure, they can appear just once), but otherwise it works. (we will hit you on that repetitiveness in *4b*; this proficiency is for the untested arbitrary-depth object composition code) (/)

2a — Your ability to handle 3D transforms is in full display here. Good job! (+)

3a — Your combined camera/first-person interaction, pre-programmed "zombie" behavior, and collision detection collectively serve as excellent demonstrations of this outcome. (+)

3e — Your vertex shader work is great here, with nice hints of what is to come in the fragment shader. Keep it up and this will work out nicely in the end. (|)

4a — Overall, your code is functional and works as intended, but the internals show multiple points of improvement. You have some code that has been commented out, and thus should eventually be removed. You diverge from some preferred JavaScript idioms, such as using `===` rather than `==`, and assigning function objects to variables rather than using the function statement (e.g., `assignVerts` in the version that I saw). Plus there is the language-independent best practice of not comparing boolean expressions to `true` or `false`. These collectively take your code's technical quality down a notch. (|)

4b — The aforementioned unnecessary repetition in your composite-object code is the main knock on your separation of concerns. One other point of improvement is to start grouping your growing number of model variables into objects. For example, it is reasonable now to put together `camera` and `zombie` objects. This will make your code easier to read, and clearly identify certain values as pertaining explicitly to specific aspects of your model. (/)

4c — This is probably the most glaring weakness in your code—boy does it look messy! The aforementioned commented-out blocks contribute to that, but in addition, you also have occasional bad (incorrect) indents and some overly long lines. A maximum of 80–120 characters is a good ballpark for line length. This truly detracts from the experience of reading your code. (/)

4d — You did a good job with working on a bunch of things on your own—zombie movement, camera (mostly), setting up walls, collision detection, keyboard codes…all are supportive of this outcome. (+)

4e — Your commit frequency is good, and your messages are descriptive. (+)

4f — You had decent beginnings of interaction by the due date, thus allowing us to consider your assignment to be submitted on time. It is also noted that you continued to steadily improve upon your work after that. (+)