CSE 167: Assignment 2 Milestone—OpenGL Scene Viewer

Ravi Ramamoorthi

Please refer to UCSD Online for the full detailed assignment 2 on the OpenGL Scene Viewer, and submit according to the standard instructions on Canvas. This document only briefly describes the milestone for homework 2; the main assignment and all the details are on UCSD Online. The end of this document also discusses extra credit opportunities. The assignment is due on Feb 14, with the milestone described here due Feb 5 (note unusual day of the week; Feb 5 is a Monday).

Homework 2 is one of the key programming assignments in the course, allowing you to accomplish a full-fledged OpenGL scene viewer. As such, you have been given **3 weeks** to do it. (This is much more generous than we have given for the timed MOOC course when we taught that, where only 2 weeks were given). However, it is an imposing assignment, certainly at least thrice as hard as homework 1, and it is very important you START EARLY. Human procrastination being what it is, we noticed that many students did not actually start until the week before. Therefore, this year, we have included a milestone, which must be submitted in the standard way (like for regular assignments) on Canvas by Feb 5. The milestone will be worth 10 points, while the regular assignment is worth 50 points, for a total of 60 points.

Milestone Requirements

To get full credit on the milestone, you will need to fully implement the transformation stack, i.e., everything except the lighting and shading. You need not implement lighting at this stage, i.e. you could shade your objects with a constant color (you might find it easier for debugging though, if you use a different color for each object, although this is not required; however, you must at least color objects with a different color from the background, so the silhouette of the scene is visible). However, everything except for lighting and shading should be implemented.

If you have implemented things correctly, you should be able to run your program on the test scenes given (i.e., run the grader program as for the complete assignment), and *the object boundaries and silhouettes should exactly match those in the solution*, even if the lighting is (obviously) incorrect. You can and should submit to the feedback servers on UCSD Online to verify this. Doing the transformation part builds in a fairly direct and step-by-step manner from homework 1, and requires relatively little nitty-gritty shader programming.

You will submit the homework 2 milestone by the due date on Feb 5, just like any other assignment, with the README file, your code, links to the feedback image and code graders etc. Obviously, your automatic grader feedback will indicate errors and large numbers of pixels off; do not worry about this, since the TAs will assign actual scores based on checking the transformations (and object silhouettes). If you have not implemented some functionality or have implemented other functionality beyond what is required, please note it it in the README file, or any other information of interest. Please note that the milestone is an important but relatively small part of your grade. It is clearly in your best interests in terms of both the score, and balancing the workload, to do as well as possible on the milestone, but a few minor errors/deductions will not substantially affect your overall project grade, if your final submission on Feb 14 is correct.

Please read the entire homework assignment on UCSD Online before starting, since this milestone is only one step in the final homework assignment. You may also want to pay careful attention to the detailed steps and hints at the end, which collect wisdom from previous (online) students on how to approach homework 2 in a step by step fashion. All of them recommend first implementing transformations, which is why we have formalized it in this milestone. As usual, please contact the instructor, graders or TAs regarding any issues, or post on Piazza.

Finally, please note that this milestone requirement is newish this year, so it is possible we may have missed some issues and we are definitely very open to addressing any unexpected circumstances. Also, if you are unable to complete everything required for the milestone, it is a relatively small part of the grade, and the majority of your score will be determined by the final submission.

Extra Credit

Historically, there has been no extra credit available for homework 2. However, we do have extra credit options for homeworks 3 and 4, and given that homework 2 is a major part of the course, we are rectifying this problem with the opportunity to earn some extra credit here (it will be a maximum of 10 points, usually significantly less; as always the effort per point for extra credit is much more than for the regular assignments).

Warning: The extra credit part of the assignment should be attempted only after you have completed the basic assignment correctly. Note that extra credit applies only to the final submission, not the milestone. As always, the amount of extra credit is likely to be small relative to the time invested, so do it only if you have fully completed the regular credit.

Motivation and Specification: The idea is to extent homework 2 into a complete scene viewer (this may require additional commands in the scene description, which you are welcome to add). We leave it to your creativity as to what features to include (credit will be somewhat subjective based on the complexity of the implementation and the writeup), but here are some options:

- *Texture Mapping:* Include support for texture mapping. This should be relatively easy since homework 0 already includes a demo of it.

- *Animation:* Include support for moving objects/animation. Again, homework 0 included a very primitive version of it.

- *Scene Graphs and Instancing:* Include better support for specifying scenes with scene graphs, encapsulating objects in an instance and re-instancing them.

- *Picking and Object-specific transforms:* You should have support for selecting individual objects and/or primitives and moving them individually in the scene (i.e., applying scale, translate, rotate to individual objects not just the global scene.

- *More complex primitives:* Include support for more complex primitives such as spline surfaces, NURBS etc. You may also want to explore geometry and tessellation shaders.

- *More complex shading:* Consider inclusion of more complex shading effects, such as shadow and environment maps, as well as more complicated reflectance models or other intricate programmable shaders. You can also consider reflection and refraction, animated textures etc. See the museum demo I showed at the start of the OpenGL lectures for some inspiration on the types of visual effects you may be able to achieve.

- *User interaction:* You can have objects do something interesting in terms of animation based on the way the user interacts with them. If pushed to the limit, this enables you to script at least a simple video game.

These are only examples, and feel free to use your creativity. The museum demo shown at the start of the OpenGL lectures is a good exemplar of what may ultimately be possible in a scene viewer, but you can also have additional user interaction and effects.

Presentation and Submission: It is critical to note that you should still submit the regular assignment, since that will still count for the bulk of the points. If you are doing the extra credit, your zip file for the source code should include an entirely separate directory with the extra credit information (this should have its own README that also explains any new scene file commands you used; you can have a global README on top identifying both the regular source code directory and the new one). Note that this submission procedure is different from homework 3 where extra credit is submitted entirely separately to its own assignment.

You will also need documentation (in the README or website) of which new features were implemented, images or videos that show that they work, and some images and videos demonstrating the types of scenes and user interaction that you create. The quality of documentation is an important part of the consideration in how much extra credit to award. We prefer that you create a website for this purpose (do not modify after the deadline) and include a link in your assignment README. But please do not post any source code on this website, or on public repositories including public github pages. Some of the credit is for showing a nice scene that showcases the visual effects with appropriate images and videos.

Finally, it may be helpful to email the TAs and instructor to let us know your submission includes extra credit, so we can watch out for it. Note that we expect only a few students to do the extra credit, so these instructions are not as well developed as for the regular assignment, but the goal is ultimately to explore your creativity and create a project you are truly proud of.