

ECS175 Project 5 Information Sheet

This is a general overview of project 5, please refer to project guideline published on SmartSite and the project prompt for additional requirements. The command syntax here is for demonstration purposes only, you don't necessarily have to follow them.

Requirements

In this project, you should use the pixel buffer to implement both a CVM and ray tracer.

Along with the requirements outlined in project guideline, here's a rough list of things we will check during grading. Make sure the following are working and you should do well. (Point value associated with specific part is in [])

1. CVM [30]: Your program should implement a full camera viewing model. The program should only have 1 GLUT window and the scene rendered on the window should be computed by various CVM parameters. The input file should be in world coordinate system and your program should implement a "world to eye" transformation. It should also allow user to change various CVM parameters and update the scene accordingly
2. Ray Tracing [50]: Each pixel on the window should be computed using the ray tracing method discussed in lecture. You can render a pixel whose ray has no intersection with any objects as black or some other default "background." You can also limit the recursion call to some constant (say 10) levels.
3. UI [15]: If your UI is functional without any issue (i.e. crashes when running some command, inconsistent with README instructions), you should receive full credit on this part. Refer to Project Guideline for some helpful resources for UI creation.
4. Manual [5]

Extra Credit

The following are published extra credit for this project.

1. Animation [5]
2. If your UI is stunning! [up to 5]

If you've done anything else extra, please document it in README file so we can assess it and try to give you more points!

Helpful Notes

When debugging the program, start with a smaller GLUT window (say 200x200) so it doesn't take too long to draw the initial scene. When the window is big, it is expected to take a long time to compute each pixel so your GLUT window might not show up for 30 seconds when launching the program.