COMPUTER GRAPHICS

ASSIGNMENT #2

SIMPLE RENDERER

INTRODUCTION

In this assignment you will continue to develop your modeling software. Your program will now render solid, shaded models.

REQUIREMENTS

Some of the requirements below involve developing some user interactions. Design it however you like, using the mouse and menu, or just the keyboard, as long as you can manipulate the scene in a reasonable amount of effort.

Your assignment is to add the following elements to your program.

1. Geometry
    - Add a uniform material for each model. The material should consist of emissive, diffuse and specular colors. Allow the user to change the different colors.
    - Add *at least* one non-uniform material (i.e. a material that is different on different vertices). There are many different options here so be creative.
    - Can be implemented after the lecture on Illumination Models.
2. The scene
    - Allow the user to add several light sources to the scene. The user should be able to position and orient the light sources, and change their colors and types. Implement *at least* point light sources and parallel light sources.
    - Add an ambient light to the scene, and allow the user to control it.
    - Can be implemented after the lecture on Illumination Models.
3. The renderer
    - Implement the z-buffer algorithm to allow hidden surface removal.
      Can be implemented after the lecture on Hidden Surface Removal.
    - Implement flat, Gouraud and Phong shading (only if normal-per-vertex are available) and allow the user to switch between them.
      Can be implemented after the lecture on Shading.

In addition, add **two** of the following:

1. Enable fog effect and supersampling anti-aliasing.
   - To add fog, you should define a fog color that is combined with the object color depending on the distance of each pixel from the viewer. This is a useful link.
   - To implement supersampling anti-aliasing, you should maintain a buffer that is larger than the screen, or in other words render an image that is larger than the screen. Then, to display it, downsample the image (using all the values, for example if your high resolution image is as twice as high and wide, compute the average of each 4 pixels) to generate the screen buffer.
2. Enable full screen blur and light bloom.
   - To implement light bloom, you should apply a gaussian filter on the bright pixels and combine it with the original image (buffer). See this link for more details.
3. Implement the clipping algorithm that was presented in tutorial 4.

SUBMISSION

Submission is frontal. Before the submission deadline, we will schedule timeslots for you to come and see us. Presentations will last 15-20 minutes, during which you will show us your work and answer our questions.

FINAL NOTES

- This is not MATAM – there is no automatic checker. This means that all the features that are to be implemented should be intuitive to the developers with plenty of room for personal interpretation. It also means that the features that you implement should behave quite differently compared two different works. Copying of any kind will not be tolerated!
- DO NOT USE any external code without permission. If you have any doubt, please ask.
- You are very much encouraged to experiment with your program and add more features to it. Previous experiences show this assignment can be addicting!
- You have until December 29th to complete this exercise, this time is more than enough, but you are strongly encouraged to start working on it right away.