CSCE 4813 – Programming Project 1
Due Date – 02/04/2019 at 11:59pm

1. Problem Statement:

The goal of this programming project is to develop a graphics program that displays images of fireworks after they have exploded. To keep things simple, we will ignore the effects of wind and gravity on the explosion, so each firework can be modeled as a collection of straight lines that all start at the explosion point, and go outwards in random directions for roughly the same distance.

Each time the "display" function in your program is called, your program should choose a random number of fireworks to display, the random (x,y,z) locations of the centers of these explosions, and the random (r,g,b) colors of the firework. Your program should then use OpenGL to draw N random lines going outward from the explosion point in random directions. For this assignment, you are not required to add smoke, or the rocket trails showing the firework being fired into the air.

2. Design:

Your first design task is to figure out how you will be generating and storing the random (x,y,z) starting points for explosions, and (r,g,b) colors for each explosion. Keep in mind that the default viewing window in OpenGL goes from [-1..1] in the X,Y,Z directions, and that colors in OpenGL can be represented either as float values between [0..1] or integer values between [0..255].

Your second design task is to figure out the starting and ending points for the lines you will be drawing. Remember that you are working in three dimensions, so each line segment will be defined by (x1,y1,z1) and (x2,y2,z2). The tricky part is to generate lines that go in all directions and are roughly the same length.

3. Implementation:

This semester we will be using C++ and OpenGL to implement all of our programming projects. The instructions for downloading and installing a Linux VM and installing OpenGL are posted in README file the "Source Code" page of the class website. Once you have OpenGL installed, you can compile your graphics program using "g++ -Wall firework.cpp -o firework -lGL -lGLU -lglut".

You are encouraged to look at sample OpenGL programs to see how the "main" function and the "display" function are normally implemented. As always, you should break the code into appropriate functions, and then add code incrementally writing comments, adding code, compiling, debugging, a little bit at a time.

Remember to use good programming style when creating your program. Choose good names for variables and constants, use proper indenting for loops and conditionals, and include clear comments in your code. Also, be sure to save backup copies of your program somewhere safe. Otherwise, you may end up retyping your whole program if something goes wrong.

4. Testing:

Test your program with different random number generator seeds until you get some images that look fun/interesting. Take a screen shot of these images to include in your project report. You may also want to show some bad/ugly images that illustrate what happens if there is a problem somewhere (e.g. too many lines, lines too short, etc.). You can discuss how you corrected these problems in your project report.

5. Documentation:

When you have completed your C++ program, write a short report using the project report template describing what the objectives were, what you did, and the status of the program. Be sure to include several output images. Finally, describe any known problems and/or your ideas on how to improve your program. Save this report to be submitted electronically via Blackboard.

6. Project Submission:

In this class, we will be using electronic project submission to make sure that all students hand their programming projects and labs on time, and to perform automatic plagiarism analysis of all programs that are submitted. When you have completed the tasks above go to Blackboard to upload your documentation (a single docx or pdf file), and all of your C++ program files. Do NOT upload an executable version of your program.

The dates on your electronic submission will be used to verify that you met the due date above. All late projects will receive reduced credit:

- 10% off if less than 1 day late,
- 20% off if less than 2 days late,
- 30% off if less than 3 days late,
- no credit if more than 3 days late.

You will receive partial credit for all programs that compile even if they do not meet all program requirements, so handing projects in on time is highly recommended.

7. Academic Honesty Statement:

Students are expected to submit their own work on all programming projects, unless group projects have been explicitly assigned. Students are NOT allowed to distribute code to each other, or copy code from another individual or website. Students ARE allowed to use any materials on the class website, or in the textbook, or ask the instructor and/or GTAs for assistance.

This course will be using highly effective program comparison software to calculate the similarity of all programs to each other, and to homework assignments from previous semesters. Please do not be tempted to plagiarize from another student.

Violations of the policies above will be reported to the Provost's office and may result in a ZERO on the programming project, an F in the class, or suspension from the university, depending on the severity of the violation and any history of prior violations.