CSC 341/csc630:Fall 2011

Assignment 1: Basic Drawing and Interaction

Assigned Tuesday, September 20. The program must be submitted to me (swei@sju.edu) by Tuesday, October 4. (any time up to midnight). Subject of the email should say “CSC341/CSC630 Assignment 1”. Name your Project directory as “YournameAssg1”. Before submitting, zip your Project directory, and email a single zipped file as attachment. Remember that your code should be fully documented. Check the course syllabus for late policy.

Overview:

In this assignment you will implement a simple 2-dimensional OpenGL program. The program should create a 400x400 window. It should assume that the idealized drawing area is a 8x8 square ranging from lower left corner (-4,-4) to upper right corner (+4,+4). The initial image should contain a blue circle on top of a red diamond, both centered in the middle of the window. The blue circle has radius 1.5. Initially it will be drawn using 100 line segments. The red diamond has height 4 and width 3. In addition, you have to draw a green equilateral triangle of side length 1.6 centered in the middle of the window, on top of other two shapes. I will leave the task of computing the vertex coordinates of the equilateral triangle as a geometric exercise. Remember from your high school geometry that the center of an equilateral triangle splits the triangle’s altitude in the ratio of 1/3:2/3.

Features:

Here is the list of features that you should implement:

- Every 10 seconds the triangle color will be toggled between green and black.
- If the left mouse button is clicked, the diamond should be drawn on top of the circle. If released, the circle should be drawn on top of the diamond.
- Whenever the window is resized, the shapes are resized in a corresponding manner.

- Pressing the ’c’ key will toggle the resolution of the circle such that it will be drawn with 20 line segments or 100 line segments.

- When the ’q’ key is hit, the program quits.

- Pressing any arrow key moves the circle in the corresponding direction by 0.1 units. (Check out the glutSpecialFunc() for further information on processing arrow and function keys.)

- SHIFT + any arrow key moves the diamond in the corresponding direction by 0.1 units. Find out which function is used to determine if the SHIFT is pressed.

- Holding down the left mouse button and dragging the mouse should translate the circle. Its center should follow the cursor. (To perform the dragging operation check out the function glutMotionFunc().) Beware that GLUT uses the upper left corner as the window origin, and OpenGL uses the “idealized coordinates” when drawing. It is your job to make the necessary coordinate transformations. That is, you will take the mouse coordinates given to you by the mouse motion callback, but you will have to map them to your idealized drawing window in order to draw the rectangle at the mouse position.

- Holding down the right mouse button and dragging the mouse should translate the diamond. Its center should follow the cursor.