Impressionistic Sampling / Rasterization – CSE 581 Lab2

Spring 2007 (Prof. Crawfis)

Objectives

- Drawing 2D primitives in OpenGL, including points and lines.
- Reading an image into a program.
- Creating a simple 2D texture map.
- Setting color, texture and vertex coordinates.
- Controlling attributes using stochastic processes.
- Understanding the basic event loop programming model.
- Developing event-based user interfaces.
- Create wonderful pieces of art.

OpenGL routines used:

- glVertex*()
- glColor*()
- glTexCoord*()
- ilutLoadImage()
- glEnable() / glDisable()
- glMatrixMode()
- glLoadIdentity()
- glOrtho() or gluOrtho2D
- glLineWidth()
- glPointSize()
- glBegin() / glEnd()

Description

Download the sample lab to use as a reference model. The core requirements are to read in an image, set it as an OpenGL texture map, randomly generate points or lines, randomly assign colors to the vertices, and set the texture coordinates either to the vertex’s position, or the line’s mid-point. User control’s to select the color, increase the linewidth or point size and to increase the total number of lines or points are required. You do not have to implement the user interface in the reference lab, but you need to document the controls substantially so that the TA can grade the lab. Notes on Random numbers and creating the Window’s Form GUI above will be on the course web page and/or posted to the course newsgroup. Note: You should only have one call to glBegin/glEnd. Do not draw each line (or point) with a separate set of these routines. Your program will be very slow. Also, do not compute the line segments and store them. This will require a fair amount of memory.

Tasks and Point Breakdown

Below is the point break-down for this lab. I have listed them in an order I would suggest you work on. Test and debug as you go.
(10 pts) Document class to encapsulate the Definition of the artwork. This would include the number of primitives, properties (thickness, etc.) of those primitives, random number generator seed point, etc.). You should be able to read and write XML files for this to save your work and re-load it. You can define your own format.

(10 pts) Draw line primitives randomly on the screen.

(10 pts) Draw point primitives randomly on the screen.

(10 pts) Change the number of primitives that are drawn under user control (up to at least 100,000).

(10 pts) Randomly assign the color for each primitive. You should have proper interpolation in RGB space.

(10 pts) Change the linewidth and/or pointsize under user control. Provide a control for the line length.

(10 pts) Restrict the random colors to lie between two user chosen colors in RGB-space.

(10 pts) Read in an image and create an OpenGL texture. I would suggest using the DevIL package for this (http://sourceforge.net/projects/openil/). At a minimum, you need to read in either JPEG or GIF images, so the TA can easily test on their image sets.

(10 pts) Assign texture coordinates for each primitive such that they coincide with the position of the point or the mid-point of the line. Enable texture mapping if a texture has been read in and initialized successfully. (You may want to provide a control to disable the texture mapping for debugging).

( 5 pts) Comment your program thoroughly.

( 5 pts) Document your program thoroughly.

Optional / Extra Credit (max 10 pts)

(5 pts) Create two of your best results and submit them with your lab for consideration on the course highlight’s web page. Please label these as: LastnameFirstname-Lab1_00.jpg and LastnameFirstname-Lab1_01.jpg. You can use a simple window capture for these (Alt-Print Screen). Use photoshop, paintshop, picture editor, Gimp, etc., to create the image file. Crop the image to an 800x600 pixel resolution. If you can, add your name and the year inside the image with a text tool. If you do not wish your name listed on the highlights page, simply pick another anonymous name.

(5 pts) Provide for semi-transparent colors and blending. Provide either a single user control to specify a fixed opacity value, or a range of opacity values.

(5 pts) Specify two texture sample locations as a ratio between the two end-points. Setting these to the same value gives a constant colored line according to the texture value at the sample point.

(5 pts) Restrict the random colors to lie between two user chosen colors in HSV-space.

(5 pts) Add a button to generate a new random seed (generate a new image).

(5 pts) Draw the original texture as a background with another user chosen color as the base color.