MyDraw - A Simple 2D Drawing Program

CS475m: Computer Graphics Minor - Assignment 1

Due Date: 21/8/2016

1 Introduction

You have to create a simple 2D drawing program that can draw lines, filled triangles and filled quads. The following classes have to be implemented:

1. A class **pen_t** that implements a pen class.
2. A class **color_t** that implements color class.
3. A class **fill_t** that implements a fill class.
4. A class **point_t** that implements a point class.
5. A class **line_t** that implements a line class.
6. A class **triangle_t** that implements a triangle class.
7. A class **drawing_t** that implements a drawing class.
8. A class **canvas_t** that implement a canvas class.

2 Class Details

The classes mentioned above must have the following features -

1. The Color Class, **color_t**
   (a) Must store the Red, Green and Blue components of a color, each component normalized between 0 and 1.
   (b) This must be used to represent current drawing color, background color, and fill colors.
2. The Pen Class, **pen_t**
   (a) A pen must have a size (how thick a line does it draw).
   (b) A pen must have a color (what color does it draw with).

(c) A pen must have a mode - this can be draw or eraser. In eraser mode, it draws with the color of the background.

3. The Fill Class, **fill_t**

   (a) Must contain the current fill color.

   (b) The fill is a solid fill, i.e., the entire fill region is filled with one color.

   (c) The actual fill algorithm that you implement is your choice, but remember this is an interactive drawing program, so you can click inside the current triangle that will be filled.

   (d) Must have a draw method that implements your fill algorithm.

4. The Point Class, **point_t**

   (a) Must contain the X and Y coordinates of a point.

   (b) Must have a draw method that plots the point at (X,Y) on the canvas. Use this to add a point to your drawing.

5. The Line Class, **line_t**

   (a) Must contain two end points of a line. These are members of the point class.

   (b) Must have a draw method that draws the line with the current color. Must use the draw method from the point class.

   (c) Implement the line drawing with the all integer Bresenham that works in all octants.

6. The Triangle Class, **triangle_t**

   (a) Must contain the vertices of the triangle.

   (b) Must contain the color of the triangle border.

   (c) Must have a draw method that draws the triangle. Must use the draw method from the line class for the boundary, and from the fill class to fill the triangle.

7. The Drawing Class, **drawing_t**

   (a) Must contain an array/list/vector of the lines and triangles that your drawing contains.

   (b) Must have a overall draw method that call the draw of all the contained elements.

8. The Canvas Class, **canvas_t**

   (a) Must contain the current drawing.

(b) Must contain the size of the canvas (width $\times$ height).

(c) Must contain a background color for the canvas.

(d) Must contain a clear method that clears the canvas.

(e) Must contain a 2D array the size of your Canvas. Each array element corresponds to a pixel on your canvas and should be able to store a color. For any point you draw to the screen color the corresponding element in the array to the same color. Assume the left bottom corner of the window to be $(0, 0)$ with the positive X-axis going from left to right and the positive Y-axis going from bottom to top of the window. To display the drawing you will directly display this array to the screen as explained below.

3 Displaying, Saving and Loading drawings

1. Read Chapter 8 of the OpenGL programming guide from here: `http://www.glprogramming.com/red/chapter08.html` to figure out how to display an array of pixels in OpenGL. Useful functions to know are `glRasterPos( ), glDrawPixels( )`.

2. Add save and load methods to the **drawing_t** class that can save the list of lines and polygons to a text file.

3. You must save all information needed to recreate your drawing from the saved file.

4. You are free to design your own file format to do this but it must be a text file format. Learn to use the C++ iostream and fstream methods to read files. Do not use C fscanf functions.

4 Drawing to be made for submission

You must make a 2D drawing of any indoor scene - your hostel room, your room at home, class room or any other building. The drawing can be from any point of view. This drawing is needed to finish the assignment.

5 Use of OpenGL and GLUT

GLUT is to be used to open your window, handle key and mouse events.

1. You must be able to open a window of a reasonable size - say 1024x768.

2. You must be able to handle the following keys and clicks in your application:

(a) 'N' : Initialize a new canvas, including all elements of the 2D array to the background color. Assume that the size of canvas is equal to the size of your window (and is fixed). Take the background color for the canvas as input from the terminal or an initial config file.

(b) 'C' : To clear the canvas with current back ground color. To do this, clear the array with the required color and draw it to the screen. This should also clear the current drawing.

(c) 'S/L' : Save/Load drawing. For load, input filename on the terminal.

(d) '1': Toggle Line drawing mode. Left clicks add points, and a line is drawn between two successively clicked points, with the current pen.

(e) '2': Toggle Triangle drawing mode. Left clicks add vertices. Three successively clicked vertices form a triangle.

(f) 'F': Fill the current triangle with the current fill color.

(g) 'G': Change current fill color. Input color from terminal.

(h) 'H': Change current pen color. Input color from terminal.

(i) 'I': Change current background color. Input color from terminal.

(j) 'J' Change current pen width. Input width from terminal.

(k) 'Esc': Exit the program.

You are allowed only **limited** OpenGL use in this assignment. You can use OpenGL for the following:

1. You will have to setup a 2D orthographic projection to setup your drawing - learn how to use the glOrtho function. For simplicity, set your coordinate frame to match the viewport size and assume integer coordinates. See the code for the Bresenham demo showed in class for example.

2. The only drawing related OpenGL commands that you are allowed to use are `glRasterPos( ), glReadPixels( )`.

3. Understand the event driven GLUT programming model before you start the assignment.

Since there is no animation involved it is ok to ask for a single buffer in glut and use glFlush at the end of the display function.

AGAIN: **No** other OpenGL functionality is to be used. Do **not** directly draw lines or polygons to the screen - that defeats the purpose of the assignment.

6 Report

1. Make a webpage. If you do not know where to host this page, see this:
   `www.cc.iitb.ac.in/engtutorials/97-how-to-create-hompage-on-homepagesiitbaci`
   The webpage must explain what you did in the assignment with appropriate screenshots for examples. Do NOT put source code on the webpage.

7 Marking

The assignment will be marked as follows:

1. Implementing the **pen_t** class correctly : 5 marks
2. Implementing the **color_t** class correctly : 5 marks
3. Implementing the **point_t** class correctly : 5 marks
4. Implementing the **canvas_t** class correctly : 15 marks
5. Implementing the **drawing_t** class correctly : 15 marks
6. Implementing the **line_t** class correctly : 15 marks
7. Implementing the **triangle_t** class correctly : 15 marks
8. Implementing the **fill_t** class correctly : 15 marks
9. Implementing the file reading and writing, loading and saving methods correctly : 15 marks
10. Implementing All the keyboard and mouse functionality : 15 marks
11. Making the example drawing : 30 marks
12. Report: 10 marks
13. Total : 160 marks
14. Late submission will follow a policy of graceful degradation with a 25% penalty for each day's delay (i.e., zero marks if the assignment is more than three days late after the due date.)
15. It is **ok** to submit a partially done assignment if you cannot complete it. You will get partial credits.
16. Please do not plagiarize source code. If you borrow from anywhere please at the source citation in a comment in your source code and also acknowledge it in your report.

8 Submission Instructions

Your submission must be in the following format:

1. A Tar-Gzipped archive of the complete source code (and only source code). It should compile using a Makefile on any Ubuntu system. Note that a .zip file is not a .tgz file. Submit a .tgz archive.

2. Include a README file in the .tgz archive containing a link to the html report page on the assignment.

3. The exact mode of submission will be discussed later in a lecture.